■パソコンユーザのための実用プログラムテクニック■

フロッピーディスク プロテクト活用ハンドブック

エリートソフト著

■基本的なプロテクトから高度なプロテクトまで分ります

■自分のソフトにプロテクトを付加できます

■ソフトの中味が分り，アレンジや修正ができます

誠文堂新光社

■パソコンユーザのための実用プログラムテクニック■

フロッピーディスク
プロテクト活用ハンドブック

エリートソフト著

■基本的なプロテクトから高度なプロテクトまで分ります

■自分のソフトにプロテクトを付加できます

■ソフトの中味が分り，アレンジや修正ができます

誠文堂新光社

目　次

フロッピーディスク プロテクト活用ハンドブック

目　次

序　文

パーソナル・コンピュータの普及と共に，年々ソフトウェアの数も増大しているが，ソフトウェアのレンタルなどにより，複製物が大量に出回るという問題が発生している。(レンタルソフト業については，1984年9月現在，ソフトウェアの著作権法が制定されていないため，その是非については，本書では特に触れない。)

このため，ソフトハウスは，自衛手段として，製品に対し，何らかの複製禁止の手段（プロテクト）を考えなければならないこととなる。

現在パーソナルコンピュータ用ソフトウェアのメディアとしては，CMT*とフロッピーディスクの2つがほとんどである。

CMTに関しては，記録方式（変調方式、ボーレート　）などがまちまちであるため，一般論としてのプロテクト法は確立が難しいが，フロッピーディスクに関しては，標準的な物理フォーマット　が確立されているため，一般論として論じやすい。

また，使用されているFDCもパーソナルコンピュータ全種類で数えるほどの種類しかなく，機種によって大きく異なるプロテクトを開発する必要がなく，開発工数の削減の点からも有利である。

もちろん，FDCの違いによるプロテクトの違いは当然あるので，これについては，第2章を参照されたい。

プロテクトに関し，プログラム開発者，及びソフトハウスとして考えなければならない点がいくつかある。

1．プロテクトをかけることにより，商品自体のパフォーマンスを落としてはならない。つまり，プロテクトのチェックに時間がかかりすぎたり，起動が不安定になってはならない。

2．プロテクトをかけた場合は，ユーザーがバックアップを作成することができないため，ユーザーの誤操作などでマスターがクラッシュした場合，クレームとなる可能性が大である。

このための何らかの対策が必要である（現行商品の中では，同内容のディスケットを予備用として付属するケースが多い)。〈注：特にOA化のためのソフト〉

以上2点が少なくとも考えなければならない点である。

これらについては，各プログラム開発者，ソフトハウスで独自のポリシーがあると考えられるのでこの点については，おまかせしたい。

プロテクトの基本的な考え方は，ターゲットマシンで使用されているディスケットの標準的物理フォーマットとは異なるフォーマットで，プログラム，データなどのリード，ライトを行なうことである。

これにより，ターゲットマシンの標準バックアップソフトでは，バックアップできないことになる。

しかし，最近は，簡単なフォーマットを変更したプロテクト程度では，ユーティリティーソフトなどでバックアップされてしまうケースが多く，さらに複雑なプロテクトが必要とされる。

本書では，この点を踏まえ，基本的なプロテクト法を紹介した後，いくつかの複合型のプロテクト法を紹介する。

複合型のプロテクト法は，その組合わせにより，無限に近い種類のプロテクト法が考えられるが，本書ではその代表例（と思われるもの）を紹介する。

読者が本書の内容をベースにして，より強力，かつ，パフォーマンスの優れたプロテクトを開発されることをぜひお願いしたい。

CMT：カセット磁気記録装置。パーソナルコンピュータの場合，ほとんどがオーディオカセットテープと同じメディアが使用され，記録，再生も，家庭用テープレコーダーが使用される場合が多い。

標準的物理フォーマット：詳しくは、1章で述べるが，たとえば，5インチディスケットの場合。MFM，40トラック／ドライブ(両面) 16セクター／トラック，256バイト／セクターが多い。

第 1 章

標準的物理フォーマット

現在，パーソナルコンピュータ用フロッピーディスクとしては，8インチ，5インチ，3.5インチ，3インチなどが使用されているが，これらは，外形こそ異なるが，基本的な磁気記録形式は，ほとんど同じものが使用されている。

この章では，プロテクトを作成するに必要な知識として，標準的物理フォーマットの解説と，これらに使用される用語の説明を行なう。

1-1　ディスケット

図1－1に5インチディスケット，8インチディスケットの外観を示す。

3インチ，3.5インチディスケットも同様な構造をしているが，5インチ，8インチディスケットよりも細部が複雑で，ジャケットもプラスチックなどで作られており，外部からの衝撃に対して強くなっている。

これは，メディアの小型化により，ユーザーの不注意によるディスケットの破壊事故が増加しないように考えられたものである。

また，プロテクトノッチの位置も異なり，さらに5インチ，8インチディスケットにはない，シャッター構造を持つ。

しかし，メディアを回転させ同心円上に情報を記録再生するといった点については何ら差異はない。

5インチディスケットと8インチディスケットの違いは，大きさの他，プロテクトノッチ，及びインデックスホールの機構が異なる点である。

5インチディスケットでは，プロテクトノッチが存在する状態で，リード／ライト可能，プロテクトシールで塞いだ状態でプロテクト状態となり，ライト動作不能となる。

しかし，8インチディスケットの場合は，逆にプロテクトノッチが存在する（つまり，プロテクトノッチが切れかかった状態）でプロテクト状態となり，プロテクトノッチが存在しない状態では，リード／ライト可能となっている。

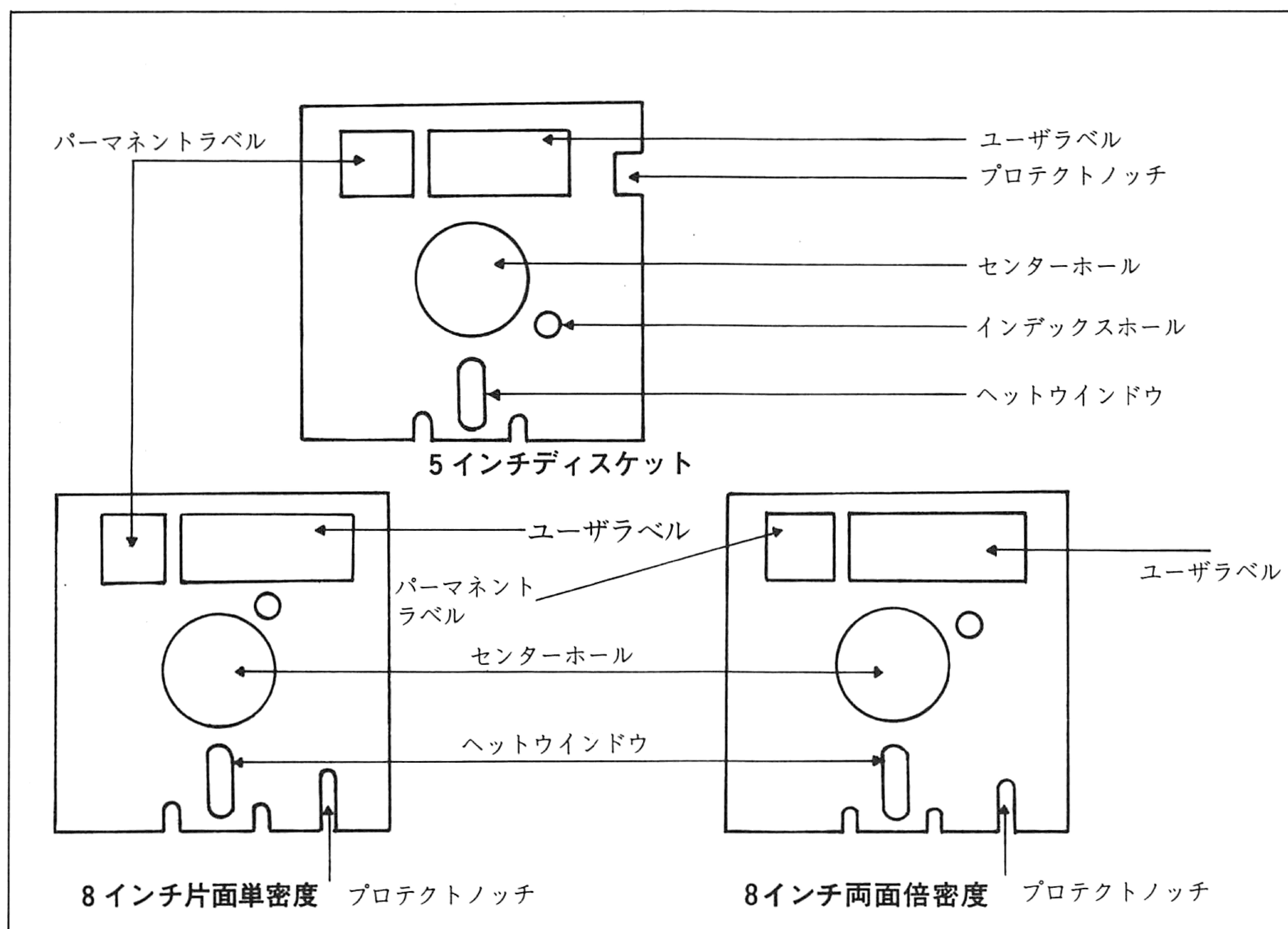
5インチディスケットは，プロテクトノッチが存在する状態で，8インチディスケットはプロテクトノッチが存在しない状態で市販されている。

5インチディスケットはメディアに書き込み後，プロテクトシールを付ければ，プロテクトされるが，8インチディスクは，メディアに書き込んだ後，プロテクトノッチの位置に相当する部分を専用の治具にて切り取らなければならない。

これは，8インチディスケットの機構上の問題点である。

このため，開発ツールなどのディスク装置には，外部から強制的にライト動作を禁止するスイッチ（ライトプロテクトスイッチ）が付いているものがある。

なお，最近のフロッピーディスクドライブは，このプロテクトノッチの検出を光学式で行なっている。



〔図1－1〕

このため，透過型の検出方式なら，プロテクトシールの材質は何でも良いが，反射式の場合は光を反射させることのできる材質でないと問題が生じる。

市販されているディスケット（5インチ）に付属されているプロテクトシールは，反射式でも問題のない材質が使用されている（一部異なるものもある）。

ただし，現在は透過式がほとんどであるので，問題が生じるのは，まれである。

インデックスホールの位置については，5インチディスケットの場合は，単密度，倍密度（これらは後述），また片面か両面かによって変化しない。片面とは，ディスケットの表側のみの面に記録する方式で，両面とは，ディスケットの表，裏両面に記録する方式である。

もちろん，記録方式が同一であれば，両面方式は片面方式の倍の容量を一枚のディスケットで持つこととなる。

8インチディスケットの場合は，片面単密度方式と両面倍密度方式では，インデックスホールの位置が異なる。どちらかと言うと，片面単密度のインデックスホールは，センターホールの真上近く，両面倍密度のインデックスホールは，真上より右に存在する。

このため，８インチのフロッピーディスクドライブは，２つの位置にインデックスホールパルス検出装置が付いている。

また，これにより，どちらの検出装置側にインデックスホールパルスが出ているかによって，片面単密度，両面倍密度いずれのディスケットがセットされているかを自動的に検出することも可能である（ただし，ソフトウェアも対応させねばならない)。

1-2 メディア

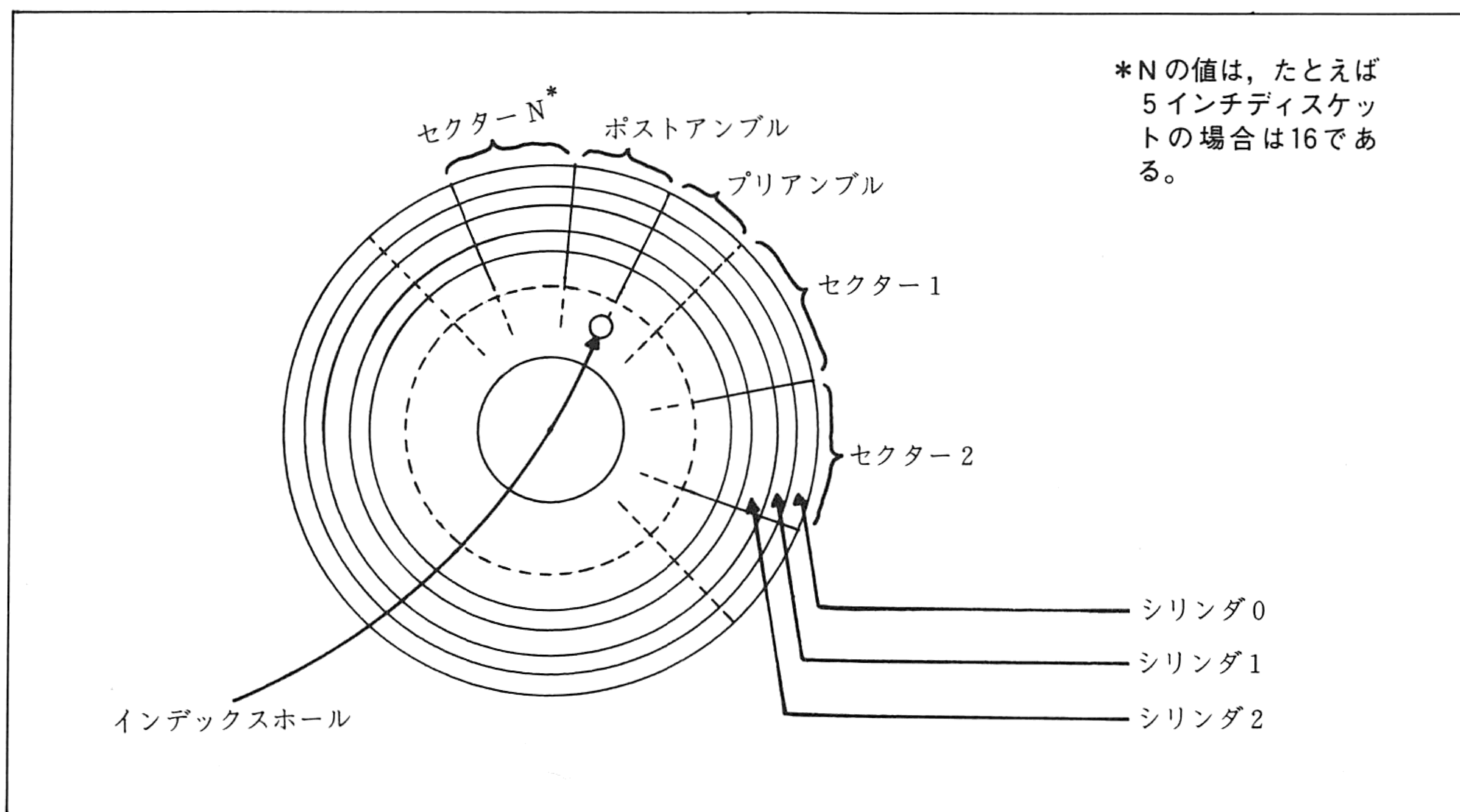
図1－2にメディア上に作成されるシリンダー，及びセクターの図を示す。

ディスケット上には，同心円方向にシリンダー，円周方向にセクターと名付けられた情報のブロックを単位としてデータなどのリード／ライト動作が行なわれる。

8インチディスケットの場合は，あらかじめシリンダー及びセクターの情報（データはダミーデータが入っている）が作成された状態（フォーマットされていると呼ぶ）で市販されているが，5インチディスケットは，フォーマットが行なわれていない状態で市販されている。

さて，シリンダー，及びセクターと共に，トラック，レコードなどの用語もよく使用され，かつ混同しやすいので解説しておく。

まず，トラックとシリンダーであるが，これらは，両方共，同心円方向の情報単位を示すが，同じ値，たとえば3であっても，実際に物理的に示される位置が異なる場合がある。



〔図1－2〕

これは，特に両面方式のディスケットの場合生ずる。

両面方式の場合，さらに，表面と裏面を示すために，ヘッド（番号）が使用され，ヘッド0が表面，ヘッド1が裏面を示す。

このヘッドとシリンダーを組み合わせて，

トラック＝(シリンダー)×２＋ヘッド

としてトラックを使用するのが一般的であるが，中には，トラック＝シリンダーであるもの（つまりヘッドは別に指定する）などもある。

いずれにせよ，使用される用語の慣例の問題となるので，ケース・バイ・ケースで使用されたい。

なお，本書では，トラック＝シリンダーとして使用しているので注意されたい。トラック＝(シリンダー)×２＋ヘッドの方を使用すると，トラック３とシリンダー３は物理的位置が等価でないため混乱すると思われるので使用しないこととした。

次にレコードとセクターであるが，これはフロッピーディスクの用語として使用される範囲では同じものである。

このように，フロッピーディスクのように，リムーバルなディスク装置でなく，一つの回転軸上にいくつもディスクが付いた大型のディスク装置に使用される用語をそのままフロッピーディスクにも使用しているため，このような不都合が生ずる場合があるので注意していただきたい。

1-3 FMとMFM

先ほど，単密度，倍密度という用語が出てきたが，これらについて解説する。

フロッピーディスクは磁気記録装置であるため，目的とする情報を何らかの磁化パターンに変換して，また逆に磁化パターンを情報に戻して，初めて，記録，再生が可能となる。

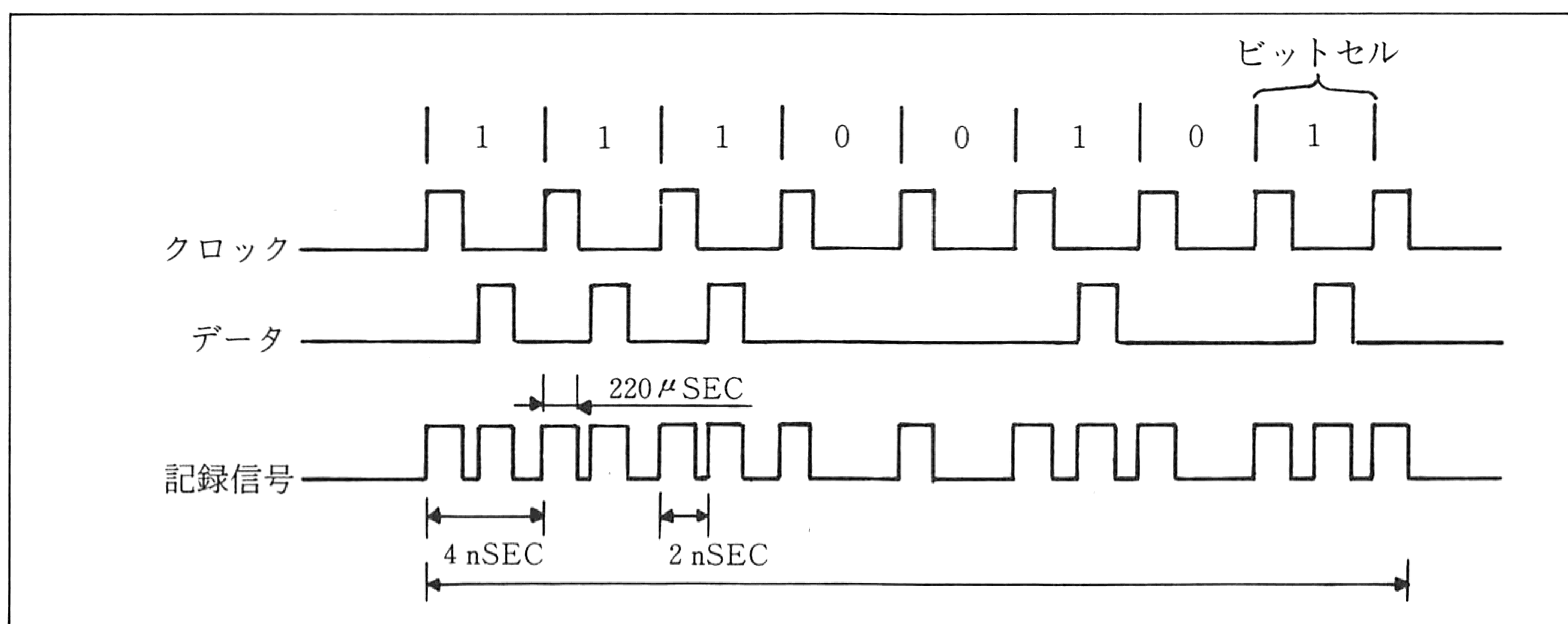
このためには，書き込み，及び読み出しの単位(時間軸方向)，及び型式を決めなければならないが，これに使用されるのがFMとMFMである。

1-3-1 FM方式

FM方式は単密度方式と同意語である。

たとえば，今，2進数化された11100101というデータを書き込む場合，実際の信号としては**図1-3**に示すような信号が書き込まれる（書き込みヘッドには，さらにこの信号に処理を行ない書き込み電流を変化させて書き込みを行なう)。

この図からわかるように，1つのビットはビットセルと呼ばれる単位で信号が作成される。



〔図1－3〕

FM 方式の規則は

1　各ビットセルの先頭にクロックビットを書く。
2　各ビットセルの中心にデータビットを書く。

である。

この図（図1 － 3）からわかるように，1バイト分のデータ長は、32μsec。使用される最大の周波数は，2 μsec＝500KHzとなる。

1-3-2　MFM方式

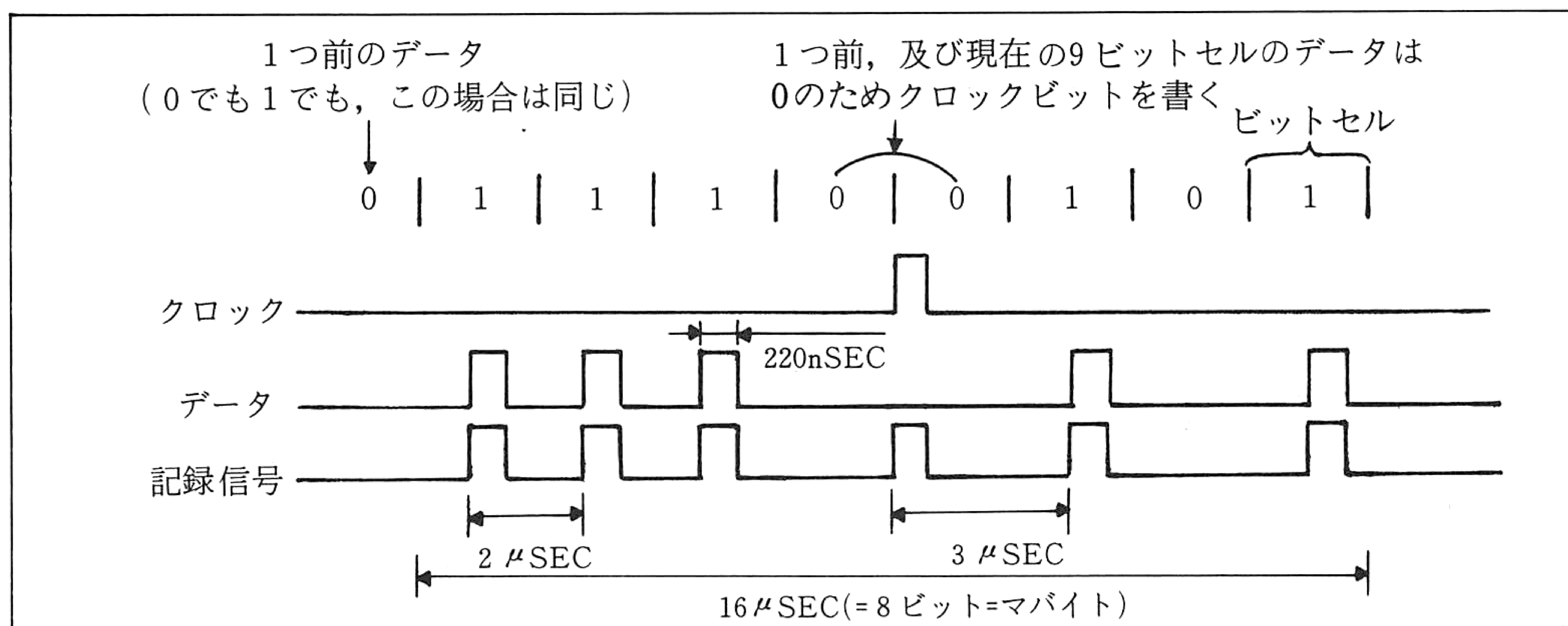
MFM 方式は倍密度方式と同意語である。

MFM のMはMODIFIED のMであり，FM 方式を一部変更したものである。

MFM 方式のビットセルは2μsec であり，FM 方式の 4μsec の半分である。

このため，記録密度は倍となるが，そのままでは，最大周波数も倍の 1MHz となってしまい，倍の周波数帯域が必要となり，アンプの性能，メディアの周波数特性などの点で問題が発生する。

このため，FM 方式の規則を一部変更して，周波数帯域はFM 方式と同じく 500KHz となるように決められている。



〔図1 － 4〕

MFM 方式の規則は,

1　各ビットセルの中心にデータビットを書く（FM 方式と同じ）
2　直前，及び現在の両ビットセルともにデータビットが 0 の場合（つまりビットが存在しない），ビットセルの先頭にクロックビットを書く。

2 の点が FM 方式と異なる点で，この方式でクロックビットを書くことにより，周波数帯域を FM 方式と同じにし，かつ記録密度を倍にすることが可能なわけである。

MFM 方式の場合,クロックビットを書くか書かないかは，1 つ前のビットセルにも影響されるので，FDC 内の処理は，より複雑である。

図 1-4 に11100101というデータを書き込む場合の記録信号を示す。

データが同じなので，FM 方式と比較していただきたい。

この図からもわかるように，最大の周波数は 2μsec＝500KHz であることが確認される。

1-4 ミッシングクロック

FM方式，MFM方式で，実際にどのような信号がメディアに記録されるのかは解説した。

いづれの方式も，メディア上には，クロックビットとデータビットから成るビットセルの羅列が書かれているわけであるが，ここで読み出し時に問題が発生する。

つまり，ちょうど，RS−232方式などのシリアルデータ転送などに使用されるデータ形式のうち，スタートビット，ストップビットに相当するものが存在しないため，どのビットセルがMSBであるのか，またLSBであるのかが判断できないことになる。

このために，ハード的にデータの同期を取る手段が必要になるが，これがミッシングクロックである。

これは，本来存在すべきクロックビットを書かないことにより，読み出し時にこれを検出してデータの同期を取るものである。

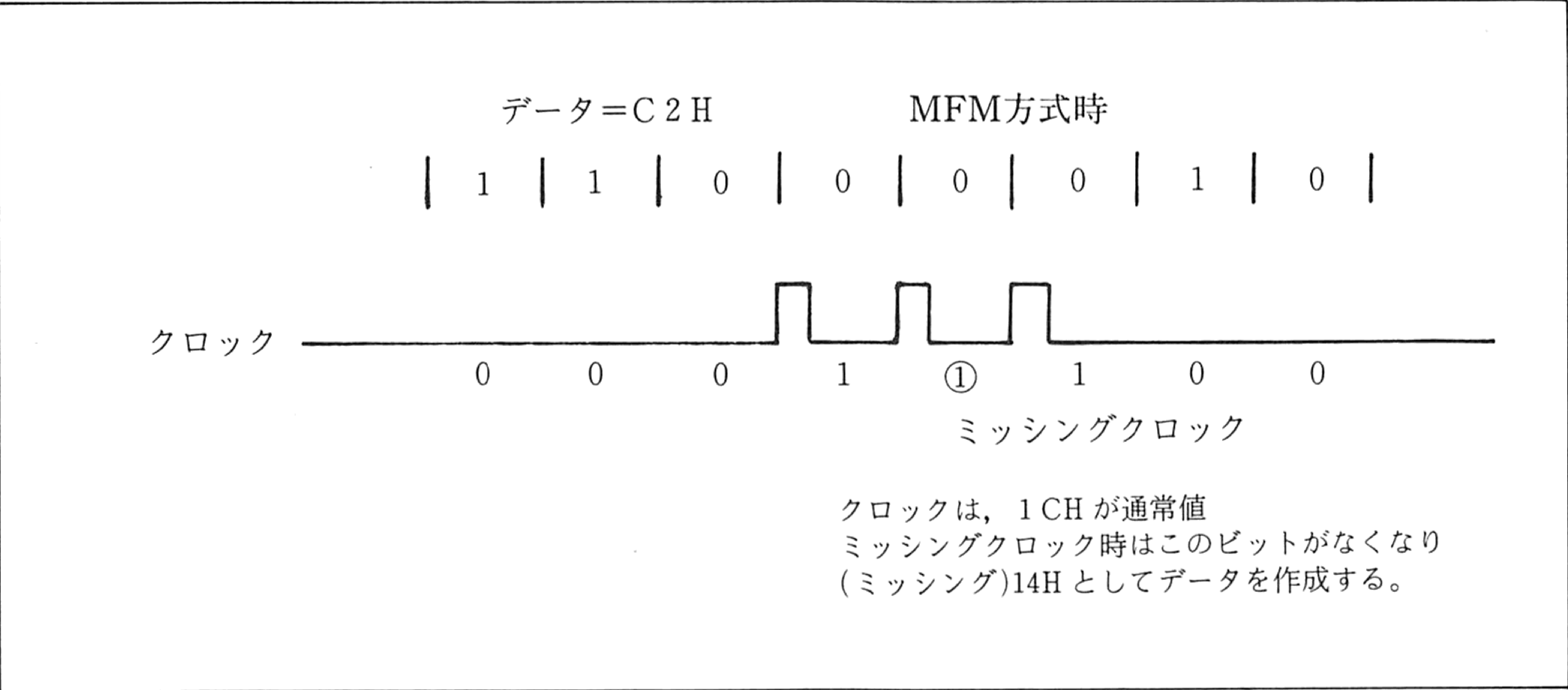
ただし，メディアが不良の場合も，あるクロックビットが存在しないために，ミッシングクロックで書かれたと判断される可能性があると考えられるが，存在しないクロックビットの位置はあらかじめ決められているので，このようなことは，まず，まれである。

図1-5にミッシング・クロックで書かれるデータとそのクロックを，また，**図1-6**に例を示す。

意　　味	データ	クロック	本来のクロック	
IAMの前に置く	C 2 H	1 4 H	1 C H	FMM方式
IDAM, DAM, DDAMの前に置く	A1 H	0 A H	0 E H	FMM方式
IAM	F C H	D7 H	F F H	FM方式
IDAM	F E H	C 7 H	F F H	FM方式
DAM	F B H	C 7 H	F F H	FM方式
DDAM	F 8 H	C 7 H	F F H	FM方式

MFM方式時は，各アドレスマークの前置3バイトのデータがミッシングクロックで書かれる。
FM方式時は，各アドレスマーク自体がミッシングクロックで書かれる。

〔図1 - 5〕



〔図１－６〕

なお，図１－５中の IAM などの用語については，後述するので参考程度に見ていただきたい。

1-5　物理フォーマット

図１－２にシリンダー（トラック）及びセクターの構造を示したが，ここでは１つのトラックがどのような構造になっているのかをさらに詳しく解説する。

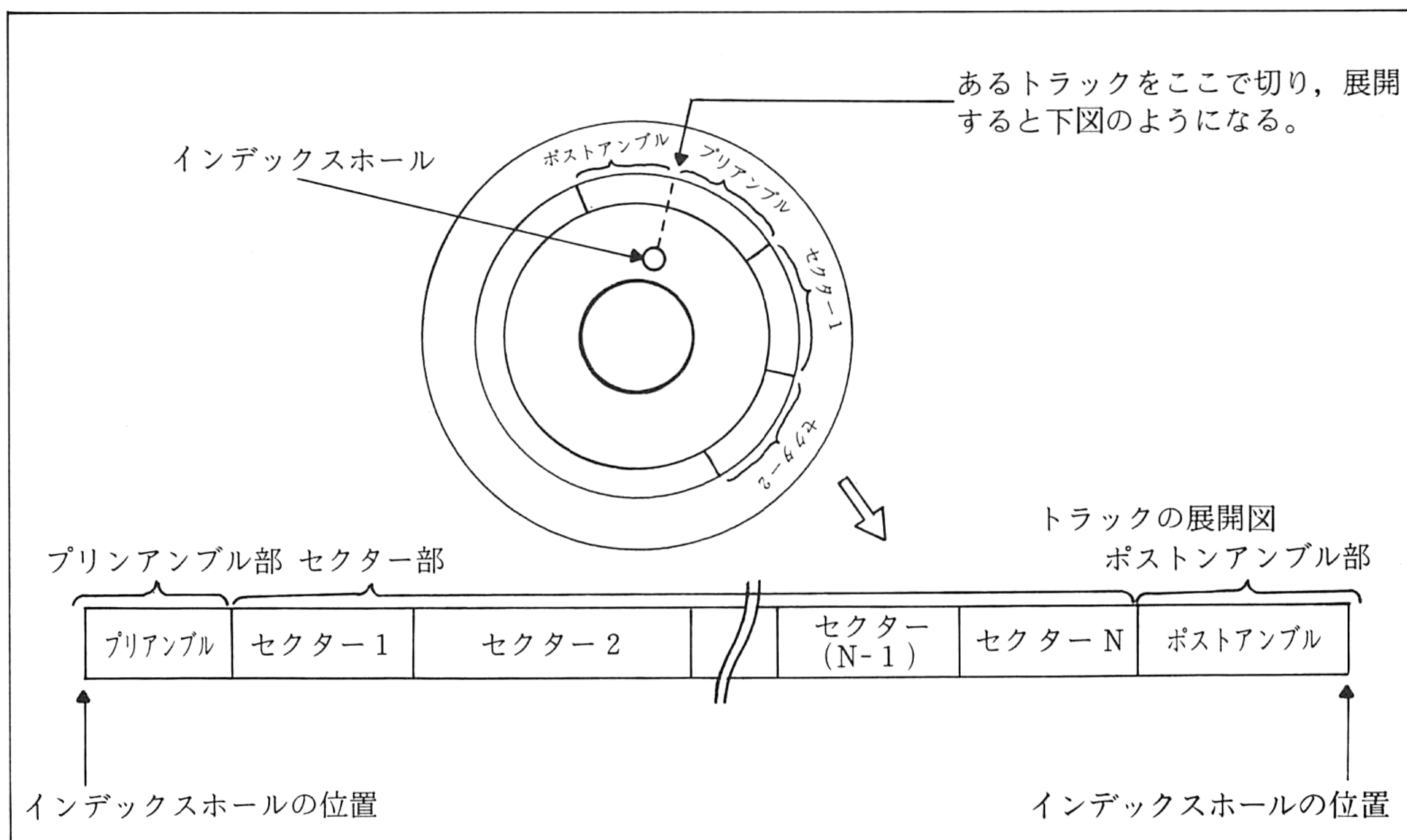
一般に，この構造を物理フォーマットと呼んでいる。

さて，**図 1-7** に１つのトラックを展開した構造図を示す。

もちろん，１つのトラックは周構造であるので，この図は，１つのトラックをインデックスホールのところで切り，展開したものである。

この図からもわかるように，１つのトラックは，

①プリアンブル部
②ポストアンブル部
③セクター部



〔図１－７〕

の3つから構成されている。

これらの構造は，互換性を持たせるために，いくつかの規格（物理フォーマット）がある．が，現在主流であるのがIBM社の提唱するフォーマットでFM方式ではIBM3740フォーマット，MFM方式ではIBMシステム34フォーマットと呼ばれている。

また，現在のほとんどのFDCがこの物理フォーマットを使用するように作られている。

次に，前述のプリアンブル部，ポストアンブル部，セクター部がIBM社の物理フォーマットではどのようになっているのかを解説する。

1-5-1　プリアンブル部

プリアンブル部は，インデックス・ホール直後に位置する部分で，ちょうど，磁気テープのヘッダーと同じ役割をはたすものである。

つまり，ディスケットやディスクドライブの精度などにより，インデックス・ホール直後から，すぐにセクター部が存在すると，最初の数バイトのデータがうまくリード／ライトが行なえない場合があるために設けられた「余白」と考えられる。

また，プリアンブル部には，この部分がインデックスホール直後であることを示すIAM（インデックス・アドレス・マーク）も含まれている。

図1-8に，プリアンブルの構成を示す。

FM方式

名称	GAP4a	SYNC	IAM	GAP1
データ（値）（16進数）	FF	00	FC	FF
バイト数（10進数）	40	6	1	26

↑
インデックスホール

MFM方式

名称	GAP4a	SYNC	IAM		GAP1
データ（値）（16進数）	4E	00	C2	FC	4E
バイト数（10進数）	80	12	3	1	50

↑
インデックスホール

上図は，たとえばMFM方式のIAMでは，実際は，C2，C2，C2，FC
のようにディスケットに書かれていることを示している。

〔図1－8〕

GAPとは，ちょうど磁気テープのショートヘッダー(データのブロックなど分けるための余白）に相当するもので，このデータ（値）そのものは何ら意味を持たず単に読み飛ばしのために使用されるものである。

プリアンブル部には，先頭のGAP4aと最後のGAP1の2つのGAPが存在し，GAP4aはインデックスホールを検出してからの読み飛ばし，GAP1はプリアンブル部を読み終わってから，次の最初のセクターを読むまでの読み飛ばしに使用される。

SYNCとIAMは，この2つ一組で意味を持ち，データの同期を取るために使用される。

つまり，ディスケット上に書かれている「0」と「1」のビットの列を，8ビットごとにうまく区切り，意味のあるデータとして読み込むために使用されるわけで，この2つが存在しないと，たとえば，

……010101010101………

というビットの列が，「AA_H」というデータの連続なのか「55_H」というデータの連続なのかが判別不能となってしまう。

厳密には，IAMのミッシングクロックによって同期が取られるのだが，SYNCとIAM 2つでデータの同期が取られると考えても間違いではない。

このIAMのアドレスマークは，必ずSYNCと組み合わせて使用され，データの同期を取る他，アドレスマーク自体の値によって，これに続くデータが何を示しているのかを示す役割も持っている。

さて，このようにプリアンブル部は，「余白」でしかないので，可能であれば存在しないほうが，この部分をセクター部に流用できるので，より容量が大きくなることは明白である。

それには，それなりにディスケットやディスクドライブの性能が向上されていなければ不可能であるが，最近では，これが十分保障されているために，プリアンブル部としてはGAP1のみしか存在しない物理フォーマットが提唱されている。

これが，ECMA（ヨーロピアン・コンピューター・マニュファクチュアズ・アソシエーション）とISO（インターナショナル・オルガニゼーション・フォー・スタンダーディゼイション）が提唱したECMA66 (ISO/TC97/SC11 N419) とECMA70 (ISO/TC97/SC11 N475)で，ECM66フォーマットはFM方式用，ECMA70フォーマットはMFM方式用である。

これらは，単純に，IBM社のフォーマットのプリアンブル部をGAP1のみにしたものと考えて良い。

1-5-2　ポストアンブル部

ポストアンブル部はプリアンブル部とは逆に，トラックの最後（つまりは，最後のセクター部の後から次のインデックスホールの位置まで）に存在する余白である。

このポストアンブルも余白であるので，この部分をセクター部に流用したほうがより容量が大きくなるが，注意しないと，最後のセクターが，はみ出てしまう場合がある。

いづれにせよ，1つのトラック内にぴったりとセクター部が入ることはないので，余白が生ずる。

これをポストアンブルと考るとわかりやすい。

ポストアンブルはGAPであり，バイト数は，セクター部の大きさによって変化する(GAPのデータ（値）そのものは他のGAPと同じで，FM時FF_H，MFM時$4E_H$である)。

1-5-3　セクター部

セクター部は，**図2-9**に示すように，さらにID部とデータ部に分かれている。

以下それぞれについて解説する。

ID部

ID部は，そのセクター部の属性などを示す指標（ID）である（図1－9）。

○ IDAM

IDAM（IDアドレスマーク）は，これ以後にセクターのID情報が続くことを示すもので，IAMと同様にSYNCと組み合わせて使用される。

○ C，H，R，N

IDAMに続くC，H，R，Nが，ID情報そのものであり，

C＝シリンダー番号（トラック番号）

H＝ヘッド番号

R＝レコード番号（セクター番号）

N＝データ部のデータ長

を示している。

たとえば，5インチ両面倍密度方式時は，

C＝0～39

H＝0または1（0が表面，1が裏面を示す）

R＝1～16

N＝1

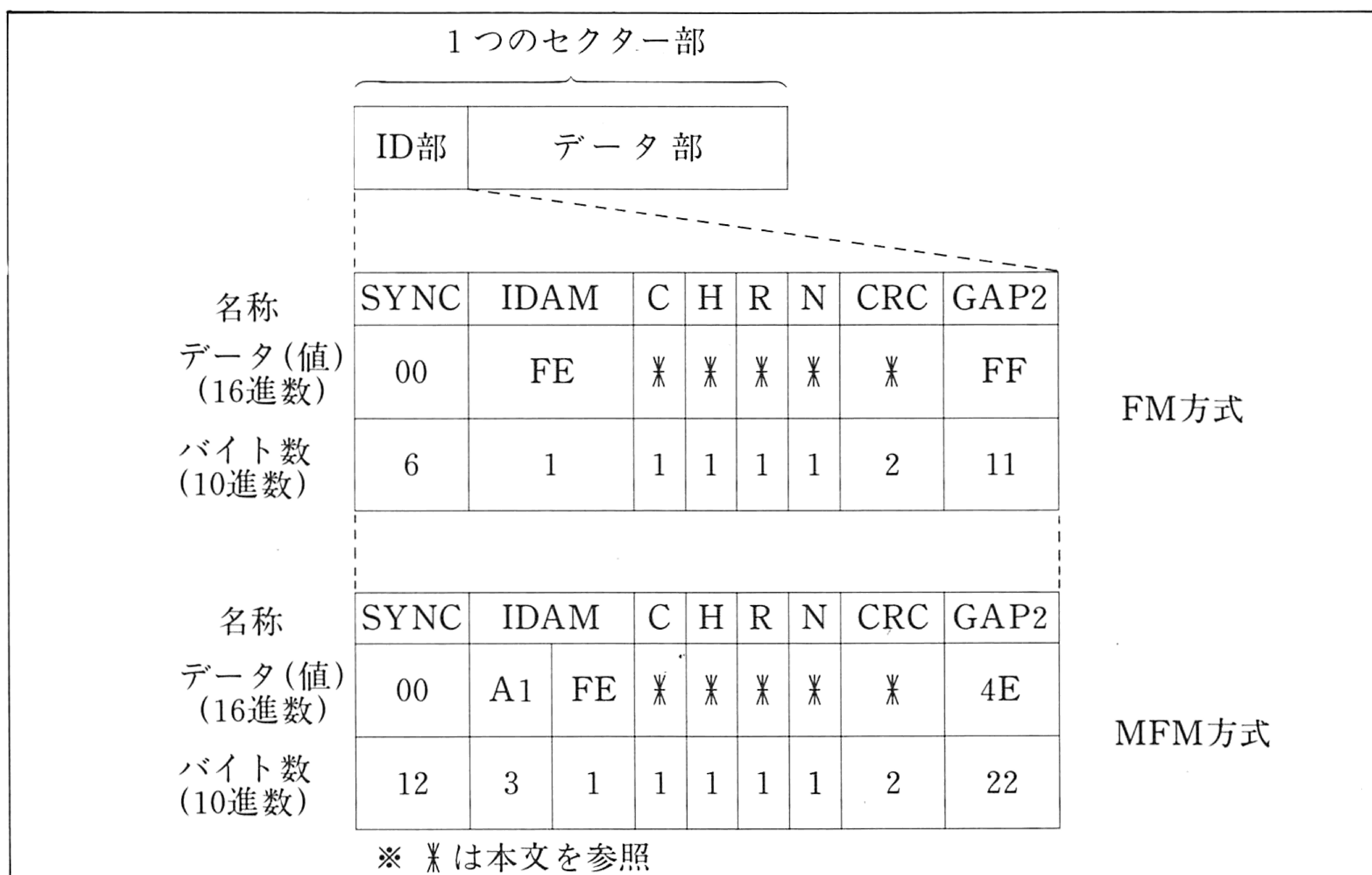
となっている。

Nはデータ部のデータ長そのものを示しているのではなく，以下のような関係になっている。

N＝0　……　128バイト／セクター

N＝1　……　256バイト／セクター

N＝2　……　512バイト／セクター



〔図1－9〕

N＝3　……　1024バイト／セクター
N＝4　……　2048バイト／セクター
N＝5　……　4096バイト／セクター
N＝6　……　8192バイト／セクター
⋮

となっている。
この表からは，

$$\text{データ長}=128\times 2^{N}$$

と考えられるが，フロッピーディスクの場合，N＝0～3程度しか使用されないため，N＝4以上の場合，この表と一致しない動作をするFDCが多いので注意を要する。
また，逆にこの点をプロテクトとして利用できるわけである。

○ GAP 2

GAP2は，ID部を読んだ後，次に続くデータ部を読むための読み飛ばし部分である。

○ CRC

CRCについては本章の最後に詳しく述べているので，そちらを参照していただきたい。

データ部

データ部は，実際にリード／ライトされるデータそのものが入っているエリアである（図1-10）。

○ DAM，DDAM

ID部と同様，データ部の先頭にはデータ部であることを示すアドレスマークが存在するが，この場合にはDAMとDDAMの2種類のアドレスマークが用意されている。
DAM（データ・アドレスマーク）は通常使用されるアドレスマークであり，DDAM（デリーテッド・データ・アドレスマーク）が使用されると，FDCによるが，該当するセクターが存在しないと判断されたり，FDCのリザルトステータス中の該当ビットがセットされたりし，DAMが使用された場合と区別されるようになっている。
もちろん，どちらのアドレスマークが使用されていても，FDCに送るコマンドのパラメー

タによりリード／ライトは可能であるので，プロテクトに利用可能である。

また，一般的にはホストマシンのDOSはDAMのみしか扱わない（DDAMの場合は，エラーと見なす場合が多い）ので，DDAMを使用するだけでもプロテクトとして使用できる。

○データ

実際に扱われるデータが入っている部分で，この長さは，ID部のNの値によって示される。

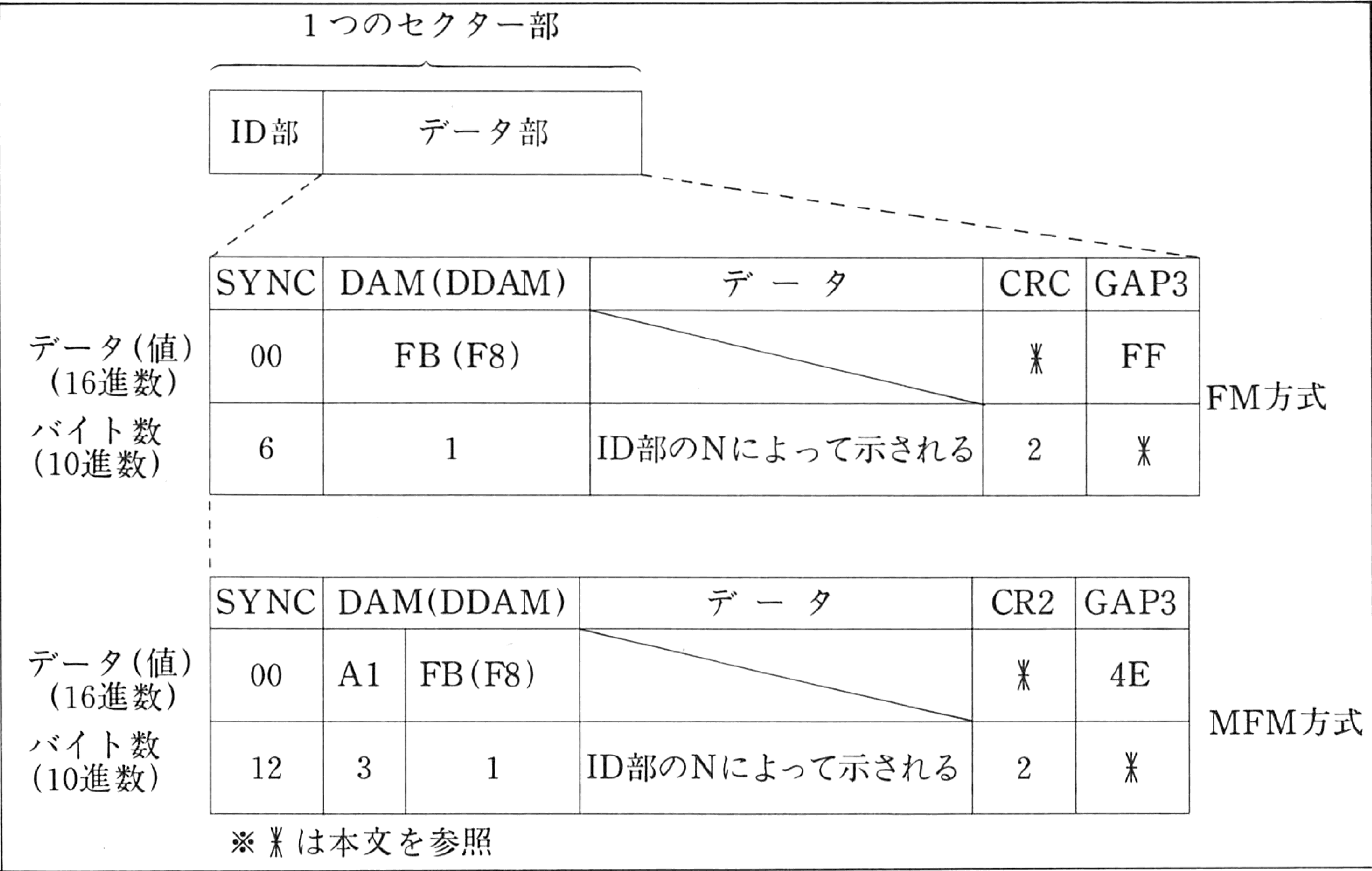
フロッピーディスクの場合，N＝1すなわち256バイト／セクターがよく使用される。

○ CRC

ID部のCRCと同じく，本章末を参照されたい。

○ GAP3

他のギャップと同じく読み飛ばしに利用されるが，このGAP3については，その長さ（バイト数）がデータ長によって変化する。



	SYNC	DAM(DDAM)	データ	CRC	GAP3
データ(値)(16進数)	00	FB (F8)		⁑	FF
バイト数(10進数)	6	1	ID部のNによって示される	2	⁑

	SYNC	DAM(DDAM)		データ	CR2	GAP3
データ(値)(16進数)	00	A1	FB(F8)		⁑	4E
バイト数(10進数)	12	3	1	ID部のNによって示される	2	⁑

〔図1－10〕

たとえば，IBM 社のフォーマットでは，次のようになる(いずれも8インチフロッピーディスクの場合)。

FM 方式時

N＝0　……　GAP3＝27バイト

N＝1　……　GAP3＝42バイト

MFM 方式時

N＝1　……　GAP3＝54バイト

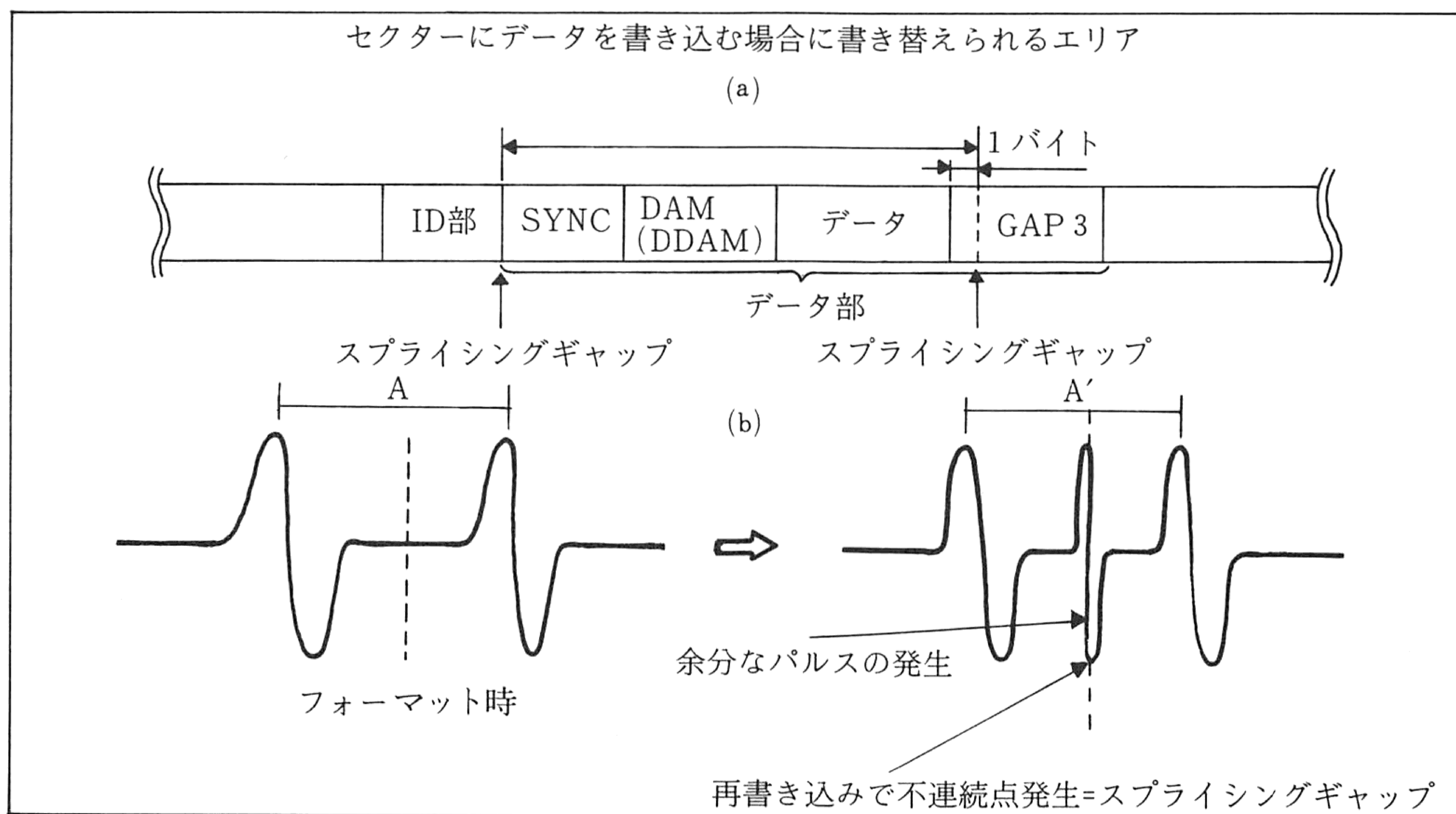
N＝3　……　GAP3＝116バイト

通常 FDC ではGAP3のバイト数を何らかのパラメータで変えられるようになっている。このGAP3の長さを利用したプロテクトも考えられる。

1-5-4　スプライシング・ギャップ

プロテクトという面からフロッピーディスクの磁気記録方式を見る場合，物理フォーマットと共に重要なのがスプライシング・ギャップの存在である。

この名称に関しては，一般に用語が規定されていないので著者が付けたものであり，本書



〔図1 - 11〕

中では，すべてこの名称を使用している。

スプライシング・ギャップとは，一言で言えば，セクターのデータ書き込み時に発生する磁気記録上の不連続点のことである。

あるセクターにデータを書き込む場合，実際に書き込みが行なわれるのは，セクター部のデータ部のみである。

実際には，**図 1-11(a)**のようなエリアに対して書き込みが行なわれるが，この書き込みは，前に書かれていたクロック，データ（最初に物理フォーマットを行なったときのもの）と完全に同期しては行なわれない。

このために，この「継ぎ目」で**図 1-11(b)**のような，不連続点（AとA′の長さ，つまり周波数のずれ，及び余分なパルスの発生）が生ずる。

これをスプライシング・ギャップと呼んでいるわけである。

このスプライシング・ギャップは，１回でも，あるセクターに対して書き込み動作を行なえば，必ず発生する。

また，このスプライシング・ギャップは，FDCのリードトラックコマンドなどで，１トラック分を読み出すと，スプライシング・ギャップの存在する場所以後のデータのビットがずれて読み出されることで知ることが可能である。

このスプライシング・ギャップを利用したプロテクトは，かなり強力である。

1-5-5　CRC

フロッピーディスクに限らず，あるメディアに書き込まれた情報が，メディアの不良（ディスケットの場合は，温度による場合が多い），または電気的なノイズなどで，誤って読み取られることがある。

読み取られたデータの正当性が不明である状態では，正確な処理を行なうことが不可能となる。

情報工学においては，このような誤りの検出，また誤りの訂正というものは，大きなテーマであり，数多くの手法が開発されている。

CRC（cyclic redundancy check）はこの誤り検出の１つの方式であり，符号化の方法がハードウェアで簡単に実現でき，かつ，復号またはチェックも容易あるために広く用いられている。

CRC方式は巡回符号（cyclic code）と呼ばれる線形符号を用いるものである。

巡回符号は，生成多項式と呼ばれるものを使用して作成される。

以下に巡回符号の原理を示す。

ディスケットに記録されたデータは，シーケンシャル（連続形式）で0と1のビットが記録されている。

この場合の0と1のビットと呼んでいるのは，クロックビットを無視し，純粋にデータビットのみを示している。

このシーケンシャルなビットの並びから，任意の数のビット数を取り出す。

たとえば，

10101001

とする。

このとき，このビットの並びを1つの式として

$$X(x)=a_n x^n+a_{(n-1)}x^{(n-1)}+\cdots\cdots+a_1 x+a_0$$

（$a_n \cdots a_0$は0または1）

の表現（これを多項式表現と呼ぶ）で表わすと，

$$x^7+x^5+x^3+1$$

となる。

この例では，ビット長は8であり，ちょうど1バイト分であるが，任意の長さで良い。

このように，巡回符号を作成する対象となる符号を，X(x)，また，巡回符号をC(x)とすると，

$$X(x)x^m=A(x)G(x)+C(x)$$

と表現される。

ここで，G(x)は生成多項式と呼ばれるものであり，mは，この生成多項式の次数を示している。

この式から，目的とする巡回符号は$X(x)x^m$を生成多項式G(x)で割った余りということがわかる。

また，A(x)は商であることも明白である。

さて，この生成多項式であるが，フロッピーディスクの場合，CCITT（国際電信電話諮問委員会）の勧告による，

$$G(x)=x^{16}+x^{12}+x^5+1$$

が使用されている。

よって，m=16となるので，巡回符号の次数は，15となり，16ビット長となる（先ほどの物理フォーマット中でもCRCが2バイトであったことを思い出していただきたい）。

さて，例として，

$$X(x) = x^5 + x^3 + x^2 + 1$$

$$G(x) = x^3 + x + 1$$

とした場合のC(x)を求めてみることにする。

この場合，系が2を法とする点（つまりmod2）と，－1は1と等価である点を考慮すれば，一般の除算と同じである。

また，m＝3から

$$X(x)x^m = x^8 + x^6 + x^5 + x^3$$

である。

よって，C(x)は，

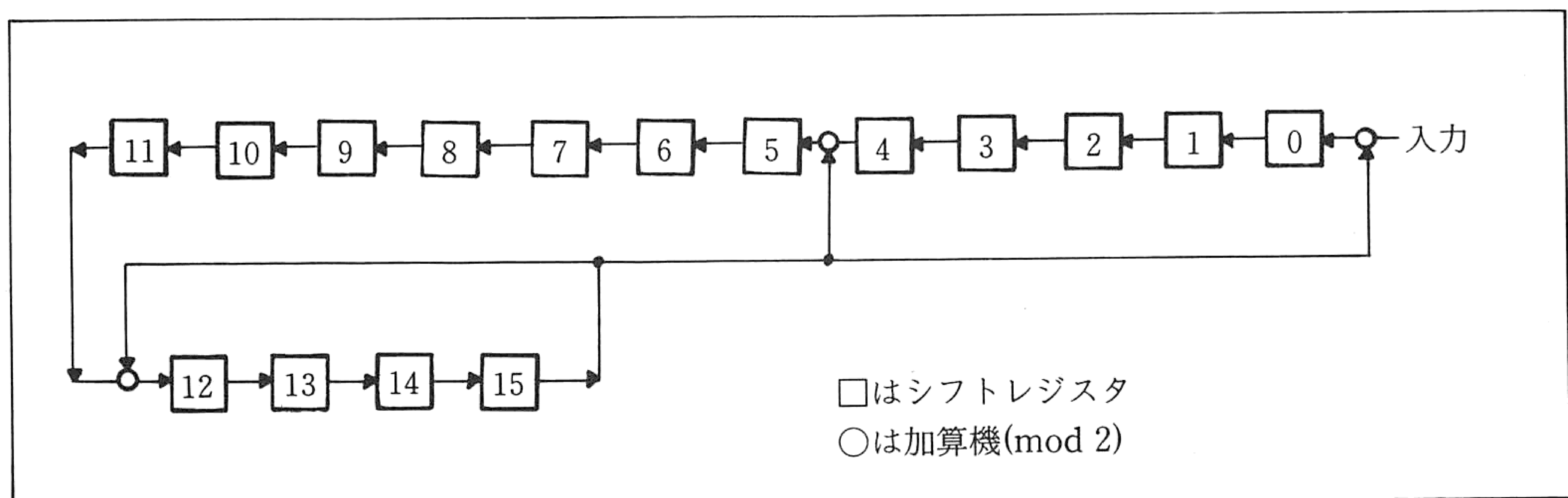
$$
\begin{array}{r}
X^5 + 1 \\
X^3 + X + 1 \;\big)\; \overline{X^8 + X^6 + X^5 + X^3} \\
\underline{X^8 + X^6 + X^5} \\
X^3 \\
\underline{X^3 + X + 1} \\
X + 1 \quad \leftarrow \begin{cases} -X \equiv X \\ -1 \equiv 1 \text{ であることに注意} \end{cases} \\
\Downarrow \\
C(x)
\end{array}
$$

から，C(x)＝x＋1となる。

このような手法で巡回符号を計算するのは，ビット長が大きくなると不利である。

このため，実際のFDC内では次に示すシフトレジスターを使用する方法が用いられている。

図1-12に巡回符号を求めるブロック図を示す。



〔図1－12〕

この場合は，生成多項式 $G(x) = x^{16} + x^{12} + x^5 + 1$ である。

右側から入ってきたデータは，クロックにシフトレジスタに取り込まれて行く。

このとき，最終段から，フィードバックが，ちょうど生成多項式の係数が1の段に入っており，ここで，2進加算が行なわれる。

すべてのデータがクロックにより取り込まれた状態のときに，現在のシフトレジスタの各段に入っているデータが，余り，つまり巡回符号となる。

どのようにして，計算されるのかは説明を略すが，実際に机上にて計算されれば，先ほどの手計算の結果と一致することが確認されるはずである。

さて，この巡回符号の計算であるが，いづれの方法でも，手で行なうのは面倒である。

しかし，プロテクトをかける場合に，あらかじめCRCがいくつになるのか知りたい場合がある。

そのため，BASICプログラムにてCRCを計算するものを作成したので，**リスト 1-1** に示す。

このプログラムは，先ほどのシフトレジスタ方式をシミュレートするもので生成多項式は $x^{16} + x^{12} + x^5 + 1$ を使用している。

もし，生成多項式を変更したい場合は，120行のM＝16を生成多項式の次数に変更し，690行の生成多項式のデータを変更すれば良い。

また，データ（X(x)）は710行に入っているので適当に変更して使用する。

RUNしてしばらくすると，結果が表示される。

なお，このプログラムは，NECのN88BASICを使用しているので他機種を使用する場合は注意が必要である（特に飛び先などにラベルを使用している点）。

〔リスト1－1〕

```
10 ' ｼﾞｭﾝｶｲ ﾌｺﾞｳ ｹｲｻﾝ ﾌﾟﾛｸﾞﾗﾑ
20 ' by ELLIE
30 ' copyright 1984 ELLIE
40 '
50 ' G(X) = ｾｲｾｲ ﾀｺｳｼｷ
60 ' D(X) = ﾌﾘｯﾌﾟ ﾌﾛｯﾌﾟ
70 ' BDAT(X) = 2ｼﾝｽｳ ｶ ｼﾀ ﾃﾞｰﾀ
80 ' M = ｾｲｾｲ ﾀｺｳｼｷ ﾉ ｼﾞｽｳ
90 ' FNA = mod 2 ﾉ ｶﾝｽｳ
100 '
110 ' initialize and definition
120 M=16
130 DIM G(M),D(M),BDAT(7)
140 FOR IL=0 TO M
150     D(IL)=0
160 NEXT
```

```
170 RESTORE *GEN
180 FOR IL=M TO 0 STEP -1
190     READ G(IL)
200 NEXT
210 DEF FNA(X,Y)=(X+Y) MOD 2
220 'main routine
230 '
240 ' read data and convert to binary
250 RESTORE *MDATA
260 READ IDAT$
270 WHILE (IDAT$<>"end")
280     IDAT=VAL("&H"+IDAT$)
290     PRINT ".";
300     FOR LP=7 TO 0 STEP -1
310             BDAT(LP)=IDAT ¥ (2^LP)
320             IDAT=IDAT MOD (2^LP)
330     NEXT
340     ' main loop
350     FOR ML=7 TO 0 STEP -1
360             DAT=BDAT(ML)
370                     GOSUB *SHIFT.REGISTER
380     NEXT
390     READ IDAT$
400 WEND
410 'add X^M's data
420 FOR I=1 TO M
430     DAT=0
440     GOSUB *SHIFT.REGISTER
450 NEXT
460 ' display result
470 RES=0:RES$=""
480 FOR LP=(M-1) TO 0 STEP -1
490     RES=RES+D(LP)*2^(LP)
500     RES$=RES$+RIGHT$(STR$(D(LP)),1)
510 NEXT
520 PRINT
530 PRINT "ｼﾞｭﾝｶｲ ﾌｺﾞｳ ﾊ ";RES$;" (";RIGHT$("000"+HEX$(RES),4);"H) ﾃﾞｽ"
540 END
550 ' simulate of shift register
560 *SHIFT.REGISTER
570 ' DAT = input data
580 D(M)=D(M-1)
590 TEMP=DAT
600 FOR LP=0 TO (M-1)
610     TEMP2=D(LP)
620     IF (G(LP)=1) AND (D(M)=1) THEN FDAT=1 ELSE FDAT=0
630     D(LP)=FNA(FDAT,TEMP)
640     TEMP=TEMP2
650 NEXT
660 RETURN
670 *GEN
680 '      16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
690 DATA    1, 0, 0, 0, 1, 0, 0,0,0,0,0,1,0,0,0,0,1  :'CCITT
700 *MDATA
710 DATA 05,1f,12,41,end    :'dummy data
```

第 2 章

基本的なプロテクト

プロテクトを考える場合，どの FDC にても，簡単にかけることができ，かつ，プロテクトのチェック方法も簡単である（逆に言えば，それだけ簡単に外されやすい）ものから FDC の能力をフルに使用し，チェックもかなりのステップが必要となるが「強力」であるものまで，さまざまである。

この章では，FDC にそれほど依存しない基本的なプロテクトについて紹介する。

2-1　IDパラメータに正常でないものを使用する

特定のトラック，セクターに対し，リード，ライトを行なう場合，その指針となるのが，ID部のIDパラメータC，H，R，Nである。

それぞれ，シリンダ番号，ヘッド番号，レコード番号，データー長を示していることは，前章で述べたが，これらの値に正常でないものを使用することによってかけるプロテクトである。

ここで正常と呼んでいるのは，ホストマシン上のDOSなどで標準的に使用されている値を意味する。

たとえば，サーフェス０，トラック番号０（最初のトラック）セクター番号５（セクター番号は１から始まる），データー長256バイト／セクターのセクターをアクセスする場合，

C＝０，H＝０，R＝５　N＝１

となる。

つまり，５インチ倍密度のドライブでは，

C＝０～39
H＝０または１
R＝１～16
N＝１

であるのが正常である。

しかし，C，H，R，の各パラメータについては，目的のトラックにシークした後であれば，単なる指針に過ぎず，どんな値であれ，書かれているIDとホストから送られるIDが一致していれば，正常にリード，ライトが実行される。

つまり，C，H，Rが，０～255のいづれの値であれ良いわけである。

たとえば，サーフェス０，トラック番号０（Cの値のことではない）のトラックのあるセクタのID値が，

C＝5

H＝3

R＝FF_H　　（$_H$は16進数を示す）

であれば，これと同じ，C，H，Rをリード，ライト時にホストから送れば，正常にリード，ライトが可能である。

この方法でプロテクトをかけると，上記のC，H，Rの値が，わかっていないと，リード，ライトできず，かつ，ホストマシンの標準DOS上では，正常なC，H，Rの値しか使用できない（していない）ので，リード，ライトが不可能となり，あたかもセクターが存在しないように判断される。

さて，N（データ長）の値であるが，これは，いささか，やっかいである。

Nの値はFDCが，リード，ライト時に実際に使用するパラメータで，

N＝0の時128バイト／セクター

N＝1の時256バイト／セクター

N＝2の時512バイト／セクター

N＝3の時1024バイト／セクター

N＝4の時2048バイト／セクター

N＝5の時4096バイト／セクター

N＝6の時8192バイト／セクター

⋮

となっており，自由に変化させることは，そのままでは，問題となる。

もちろん，イニシャライズ時に，512バイト／セクターでフォーマットされているセクターに対して，N＝2の値で，リード，ライトをするには，問題がないが，256バイト／セクターでフォーマットされているセクターにN＝2でライトする（もちろん，書かれているNの値も2でなければならない）と，次のセクターのID部以後までデータが書き込まれてしまい，次のセクターが，見かけ上，消えてしまうこととなる。

逆に，512バイト／セクターでフォーマットされている，セクターに対し，N＝1でリードすると，データ部にCRCエラーが発生する。

実は，より強力なプロテクト（次章参照）は，逆にこれらの事実を利用しているのであるが，この章では扱わないこととする。

以上から，Nの値については，C，H，Rの値とは異なり，フォーマットしたときのデータ長と同一とするのが基本的な方法である。

しかし，一般的には，256バイト／セクター（N＝1）が使用されているので，フォーマット時に，この値以外の，512バイト／セクター，128バイト／セクターを使用（もちろん，ID部に書き込む，Nの値も，それぞれN＝2，N＝0とせねばならない）するだけでも，かなりのプロテクトとなる。

また，Nの値の扱いについては，FDCに作用されることがあり，NECのμPD765シリーズでは，N＝0～6の範囲で，先ほどのデータ長が保障されているが，富士通のMB8876シリーズでは，Nの値の下位2ビットしか使用されない(それ以上のビットは，マスクされる)ので，N＝1，N＝5，N＝9，N＝13は，すべて，N＝1，つまり，256バイト／セクターとして処理される。

よって，N＝4以上の場合のリード，ライトは両者で異なることを注意されたい（逆にこの違いは，第5章で有効なプロテクト法となる)。

さらに，フォーマット時に，μPD765シリーズでは，WRITE IDコマンドにより，ホスト

```
B
ID dump program
drive No.? 1
track No.? 0
How many sectors? 16
NO.         C         H         R         N
 1          0         0         1         1
 2          0         0         2         1
 3          0         0         3         1
 4          0         0         4         1
 5          0         0         5         1
 6          0         0         6         1
 7          0         0         7         1
 8          0         0         8         1
 9          0         0         9         1
 10         0         0         A         1
 11         0         0         B         1
 12         0         0         C         1
 13         0         0         D         1
 14         0         0         E         1
 15         0         0         F         1
 16         0         0         10        1
Hard copy? y
```

〔図2-1〕-その1　正常なID

から自由なC，H，R，Nの値（00H～FFH）を送れるが，MB8876シリーズでは，ライトトラックコマンドによりフォーマットを行なうため，MFMモード時には，F5H～F8H，FBH，FCH，FEHが特別な意味を持つ。

このため，C，H，R，Nの値に上記のデータは使用できないので注意していただきたい（これらの値を使用すると，内部で計算されたCRCや，ミッシングクロックでデータが書かれたりする）。

これを利用したプロテクト法も，第4章で紹介する。

以上の手順をまとめると，以下のようになる。

1．あらかじめ決めておいた，C，H，R，Nの値，かつNのデータ長で，フォーマットを行なう。

2．1．のC，H，R，Nを使用して，データの書き込みを行なう（必ずしも書かなくとも良い）。

以上で，プロテクトがかかる。

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 16
NO.             C              H              R              N
 1              23             12             64             1
 2              84             62             12             1
 3              F6             2D             6E             1
 4              D6             8E             44             1
 5              46             84             26             1
 6              65             48             95             1
 7              11             FE             65             1
 8              E9             F5             D6             1
 9              49             E5             F4             1
 10             6E             2D             FA             1
 11             6D             2C             6E             1
 12             D8             E3             2C             1
 13             E8             D6             FD             1
 14             F6             E5             D6             1
 15             C9             EF             DF             1
 16             56             E8             F4             1
Hard copy? y
```

〔図2-1〕-その2　プロテクトをかけたID(例)

プロテクトのチェック方法は，

1．フォーマット時に使用したC，H，R，Nでデータをリード，ライトし，エラー（IDが見つからないなど）となった場合，マスターディスケットではないと判断する。

2．リードIDコマンドを使用して，フォーマットした時のC，H，R，Nが，すべてのセクターについて一致しているかチェックする。

3．同じく，リードIDコマンドにて，フォーマットした時のC，H，R，N以外のセクターが存在するかチェックする。

などである。

図2-1に正常なIDとプロテクトをかけたIDの例を示す。

2-2 FMフォーマットを使用する。

ディスクの大容量化に伴い，単密度(FM フォーマット)が使用されることは，パーソナル・コンピュータでは，ほとんどないと言って良い。

この点を逆に利用し，あるトラック，または全トラックに対して，FM フォーマットを使用すれば，それだけでホストマシンの標準DOS などでは，リード，ライトが不可能であるため，有効なプロテクトとなる。

しかし，逆に，1トラックあたりの容量が，単純に考えて，約半分となるので，パフォーマンスは悪くなる。

具体的な方法としては，フォーマット時，及び，リード，ライト時に，FM フォーマットで行なえば良いだけなので，簡単である。

μPD765 シリーズの場合，コマンド中の MF ビット（MFM モードであることを示すビット）を 0 にすれば良く，MB8876 シリーズの場合は，$\overline{\text{DDEN}}$（ダブルデンシティー）の端子を“H”（5 V）としてやれば良い。

μPD765 シリーズでは，FM/MFM の選択がコマンドにて指定できるが，MB8876 シリーズの場合，LSI の端子に FM/MFM の選択が設けられている。

このため，ハード的に，MFM に固定されているパーソナル・コンピュータがあり，このような機種では，FM フォーマットを使用することは不可能である。

2-2-1 DDAMを使用する

データ部のアドレス・マークには通常 DAM が使用されるが，DAM の代りに DDAM を使用して，データを書き込むプロテクトである。

DDAM を使用したデータが書き込まれたセクターに対しリードを行なうと，実行後のステータスが，DAM の場合と異なる結果となる (DDAM で書かれていることを示す何らかのビットが変化するが，これは FDC によって異なる)。

このため，DAM であることを前提にしてホストから，そのセクターがリードされると，

DDAMで書かれているため正常にリードできなかったと判断されるわけである。

しかし，ステータスが変化しても，適当なコマンドパラメータをFDCに送れば，データ自体は，読むことが可能であるので，DAMを使用した場合と同じように使用できる。

なお，DDAMは，ライト時に書き込まれるので，1トラック中に，DAMのセクターとDDAMのセクターを混在させると，より効果的である。

2-3 アンダーセクター法

通常，５インチ，MFM，N＝１の場合，１トラックのセクター数は，16であるが，これより小さいセクター数にてフォーマットするプロテクトである。

たとえば，８セクターとしてフォーマットを行なった場合残り８セクター分のエリアには，すべてギャップが書き込まれる。

この方法でプロテクトをかけたトラックに対し，IDを読み出し，８セクター分以上のIDが発見されれば，マスターディスケットではないことが確認される。

2-4 オーバーセクター法

アンダーセクター法とは逆に，セクター数を標準値より大きくするプロテクトである。

たとえば，前述の例で，セクター数を18として，フォーマットを行なう。

チェック法は，アンダーセクター法と同じく，IDを読み出し，18セクター分のIDが発見されなければ，マスターディスケットではないと判断する。

このプロテクトの特徴は，標準のホストマシンのユーティリティー(バックアップソフト)では，16セクター分のセクターしかコピーされず，よってエラーが発見されないため，一見プロテクトが，かかっていないように考えられる点にある。

さて，オーバーセクター法の場合，1点注意しなければならないことがある。

それは，容量の決っているトラック内に，標準値以上のセクターを作成するために，そのままでは，セクターが「はみ出す」問題が発生する。

このため，各セクターのGAP3の長さを短くして，1トラック内におさまるようにしなければならない。

GAP3の長さは，FDCのコマンドパラメータとして指定できるもの（μPD765シリーズ）と，ホストからのデータのバイト数によって指定できるもの（MB8876シリーズ）などがある。

GAP3の長さをあまり短かくすると，リード，ライトに問題が発生する場合がある。

2-5 アンフォーマット・トラック法

あるトラックに対してフォーマットをしないことを利用したプロテクトである。

8インチディスケットの場合は，あらかじめIBMフォーマットにてフォーマットがかけられているが，5インチディスクは，特殊なものを除いては，フォーマットがされていない。

8インチディスケットは，すでにフォーマットがかけられているため，アンフォーマットに戻すには，外部から，磁気を加えて，消去する方法が考えられるが，専用の治具などがないと危険である。

そこで，ソフト的に消去することを考える。

アンフォーマットと言うのは，極論すれば，アンドレスマークが1つも存在しないことである。

つまり，FDCは，リード，ライト時に，アドレスマークを見つけてから次の処理を行なうので，アドレスマークが存在しなければ見かけ上，アンフォーマットの状態と同じになる。

よって，FDCにて，そのようなアンフォーマットのフォーマットをかければ良いことになる。

このやり方は，FDCによって異なるので，個別に説明する。

1. μPD765シリーズの場合

μPD765シリーズを使用して，フォーマットを行なう場合，WRITE IDコマンドを使用する。

しかし，単にこのコマンドを使用したのでは，アドレスマークは書き込まれてしまう。

ところが，μPD765で，1トラックの容量を越えるようなフォーマットを行なうと，次のインデックスホールを無視し，2周目の最後まで書かれてしまうといった事実がある。

たとえば，5インチ，MFM，N＝6でフォーマットを行なうと，**図2-2**のようになる。

このようにFDCは，見かけ上トラック2周分を1つのトラックと見て処理を行なうので，1周目に書いたプリアンブル部やID部，データ部先頭のアドレスマークなどは，2周目の書き込みによってすべて消されてしまい，アドレスマークが存在しなくなる。

よって，アンフォーマットと等価になるわけである。

2. MB8876 シリーズの場合

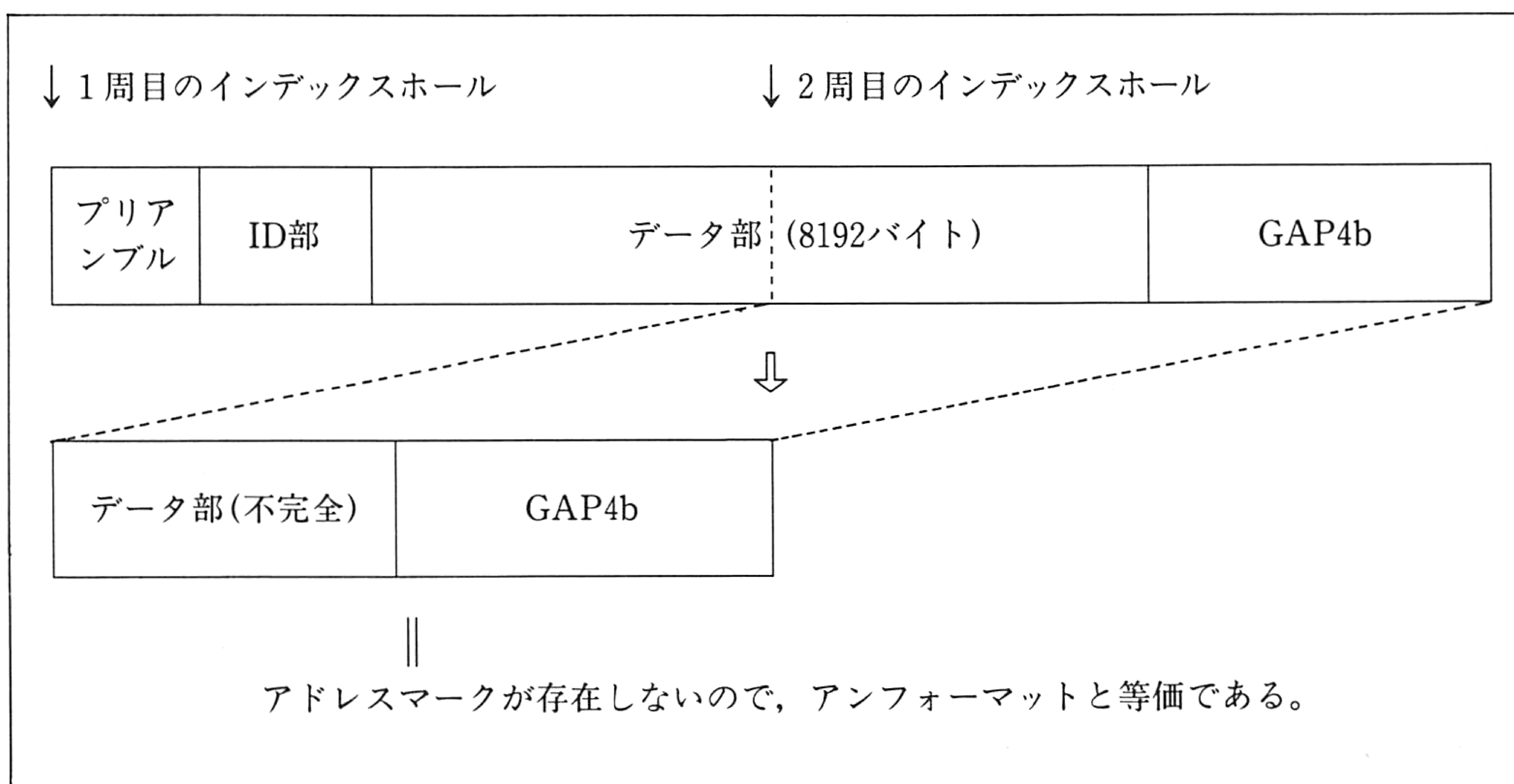
MB8876シリーズの場合は，ライト・トラックコマンドによってフォーマットを行なうが，実際は，プリアンブル部，ID部，データ部，ポストアンブル部の全初期データをホストから送ることによってフォーマットを行なっている。

よって，これらのデータパターンを変更してやれば，IBMフォーマットやECMAフォーマット以外のフォーマットも可能であり，アドレスマークを含まないフォーマットを行なうことも可能である。

一番簡単な方法では，ライト・トラックコマンドを実行し，最初の1バイトのデータ（ただし，00H～F4H）をDRレジスタに書き込めば，以後はすべて00Hのデータが1トラック分書き込まれるため，アドレスマークが存在しなくなる，というものである。

注意しなければならないのは，MB8876シリーズの場合逆にリード・トラックと言うコマンドがあり，このコマンドでは，フォーマットによらず1トラックの全データを読み出すことが可能であるので，上記の方法だと，データが読み出されてしまい，本来のアンフォーマットと区別されてしまう点である。

この場合は，リードIDコマンドを実行し，有効なIDが存在しないことで，アンフォーマットであると判断するのが適当であろう。



〔図2-2〕μPD765で，N=6でフォーマットする。

2-6 オーバートラック法

5インチドライブ（倍密度）の場合，サーフェスを別と考えれば，使用されているトラックは，0から39までであるが，39トラック以上のトラックを使用することによってかけるプロテクトである。

単純には，39トラック以上のトラックに対し，フォーマッティングを行ない，リード，ライトをすれば良いわけである。

しかし，この方法は，ドライブ及びディスケットのスペックを越えたトラックを使用するので危険が多く，お勧めできない。

また，ドライブによっては，39トラック以上にシークできないものや，ディスケットによっては，39トラック以上の部分にカレンダー処理（鏡面処理）が行なわれていないために，この部分をアクセスすると，ヘッドが損傷する場合もあるので，実験レベルの場合にも注意していただきたい。

2-7 ライトできないことを利用する

ビジネス・ソフト，ユーティリティーソフトなどでは，マスターディスケット保護のためから，ライトプロテクトシールを張り付けたり，プロテクトノッチのないディスケットを使用するのが一般的である。

これは逆に，ある使用してないセクターに対して，ライト動作を行ない，ライトプロテクトエラーとなれば，マスターディスケットであると判断することにも使用できる。

ただし，マスターディスケットであっても，ユーザーがライトプロテクトシールを剥すと問題であるので，剥せない構造にしておく必要があり，逆に，コピーされたディスケットにも，プロテクトシールが張られると，マスターディスケットであると判断するので，他のプロテクトと併用するのが望ましい。

2-8　8インチディスク特有のプロテクト

以上のプロテクトはメディアに関係なく使用できるものであるが，8インチディスクの場合，さらに特有のプロテクトをかけることが可能である。

まず，5インチの場合は片面，両面及び単密度，倍密度に係わらず，メディアの外観は同じであるが，8インチの場合は，片面単密度と両面倍密度とでは，インデックスホールの位置が異なる。

このため，8インチは，書き込まれているデータを読み出さなくとも，ドライブからのインデックスホールの位置情報により，片面，両面，あるいは単密度，倍密度を判断することが可能である。

そこで，これを逆に利用し，インデックスホールの位置情報とは異なる密度などでフォーマットを行なうことにより，プロテクトをかけることが可能となる。

たとえば，片面単密度のディスケットに両面倍密度のフォーマットを行ない使用する。

しかし，メディアのインデックスホールの位置情報，たとえばμPD765のリザルトステータス3（ST3）で読み取られるTWO SIDEビットは，片面単密度のそれでなければならないので，もし，一致しなければ，マスターディスケットではないと判断できる。

このプロテクトは案外気が付かないので，有効であるが，メディアの信頼性（単密度のディスケットを倍密度で使用するため）及び，インデックスホールの位置情報が，何らかの形で読み取れなければならない点に注意を用する。

次に両面倍密度の場合は，トラック0，サイド0.1の両方が単密度でフォーマットされているので，これを利用するプロテクトが考えられる。

一番目の方法は，この単密度のトラックを倍密度にて再フォーマットして利用する方法である。

この場合，特にホストマシンがこのトラックを使用していない場合に効果がある。

2番目の方法は，トラック0，サイド0.1のみを倍密度にてフォーマットを行ない，残りのトラックをすべて単密度にてフォーマットを行ない使用する方法である。これは前述の単密度のディスケットに倍密度のフォーマットを行なって使用する方法の逆である。

上記のように，メディア，もしくは，ホストマシンの特長を良く知ることによって特有のプロテクトもかけることが出来ることを知っていただきたい。

第3章

より高度なプロテクト

この章で紹介するのは，FDC ごとの特徴，能力をフルに使用した，比較的高度なプロテクト法である。

ここでは，現在市場でかなりのシェアを得ている NEC の μPD765 シリーズ（以下765と略す）と富士通の MB8876 シリーズ（以下8876と略す）についての各種プロテクト法について解説するが，同じプロテクト法でも，FDC により，かけ方，チェック法が異なる場合もあり，これについては，別々に解説する。

また，各プロテクト見出しの下に対応する FDC が入っているので，利用されたい。

3-1 IDの並びに関するプロテクト

■665，8876

第2章では，ID部のC，H，R，の値を変化させることによってかけるプロテクトについて解説したが，ここでは，さらにIDの並び方自体を変化させることによってかけるプロテクトについて解説する。

このプロテクトは，かけることよりも，チェックに少々手間がかかる。

図3-1は，それぞれ，1トラックの全セクターIDを読み出した表である。

No.1からNo.16までのセクターが存在しているが，No.1のセクターは，インデックスホール後最初のセクター，No.16は，最後のセクターを示している。

Bの方は，一見，C，H，Rの値を変化させたプロテクトをかけたように見えるが，よく見ると，IDの並び方が，まちまちなだけで，すべて，正常なC，H，Rが使用されている。

よって，Bでも，リード，ライト動作は正常なフォーマットAとまったく同様に可能となる。

このため，単なるリード，ライトでチェックしたのでは，マスターディスケットか否かの判断はつかない。

そこで，IDを連続的に読み出し，IDの並びがプロテクトをかけた並びと一致するかどうかのチェックを行なう。

765，8876共に，リードID，リードアドレスのコマンドを有しているので，これを利用するわけである。

注意点としては，上記のコマンドは，インデックスホール後最初からのIDを読むのではなく，コマンド起動後，最初に発見したIDを読み取るので，隣り合う2つのセクターID間の並びはチェックできるが，トラック上のIDの絶対位置は認識できないことである。

絶対位置を認識するプロテクトを次に紹介する。

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 16
NO.        C        H        R        N
 1         0        0        1        1
 2         0        0        2        1
 3         0        0        3        1
 4         0        0        4        1
 5         0        0        5        1
 6         0        0        6        1
 7         0        0        7        1
 8         0        0        8        1
 9         0        0        9        1
 10        0        0        A        1
 11        0        0        B        1
 12        0        0        C        1
 13        0        0        D        1
 14        0        0        E        1
 15        0        0        F        1
 16        0        0        10       1
Hard copy? y
```

(A) 正常なIDの並び

(B) プロテクトをかけたIDの並び(例)(インターリーブ6フォーマット)

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 16
NO.        C        H        R        N
 1         0        0        1        1
 2         0        0        E        1
 3         0        0        B        1
 4         0        0        8        1
 5         0        0        5        1
 6         0        0        2        1
 7         0        0        F        1
 8         0        0        C        1
 9         0        0        9        1
 10        0        0        6        1
 11        0        0        3        1
 12        0        0        10       1
 13        0        0        D        1
 14        0        0        A        1
 15        0        0        7        1
 16        0        0        4        1
Hard copy? y
```

〔図3-1〕

3-2 IDの絶対位置に関するプロテクト

■765, 8876

IDの並びをチェックするプロテクトより強力なプロテクトである。

図3-2，Bを見ると，この並び方では，前述のIDの並びをチェックするプロテクトでは，AとBが同一と認識される。

このため，インデックスホール後，最初のIDを読み出し，その後，それ以後のセクターIDを，読み出せばチェックが可能となる。

以下に，この手順を示す。

(1)765の場合

765の場合，リード・ダイアグノスティックコマンドを使用して，インデックス・ホール直後のセクターをリードできるので，これを利用する。

リード・ダイアグノスティックは，リード・データコマンドと異なり，インデックス後最初のセクターのIDとホストマシンから送られたIDとを比較し，異なっていても，リード処理は行なわれる。

しかし，ID部のIDとホストマシンからのIDが異なっていた場合，ST1(リザルトステータス1)のD2ビットが立つので，最初のセクターのIDが一致しているかどうかのチェックが可能となる。

Bの場合，C＝0，H＝0，R＝3，N＝1にて，リードダイアグノスティックコマンドを実行後，ST1のD2ビットが立っていなければ，マスターディスケットと判断できる。

この後，さらに，リードIDコマンドにて，1トラック分のセクターを読み出し，プロテクトをかけたIDとすべて一致しているかどうか，チェックをすれば，良い。

なお，上記のリードダイアグノスティック実行後，即リードIDを行なえば，2番目のセクター以後が読み出せるので，これを基にテーブルを作り，チェックを行なっても良い。(図4－2のIDダンプのプログラムは，これを利用している)。

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 16
NO.            C              H              R              N
 1             0              0              1              1
 2             0              0              2              1
 3             0              0              3              1
 4             0              0              4              1
 5             0              0              5              1
 6             0              0              6              1
 7             0              0              7              1
 8             0              0              8              1
 9             0              0              9              1
 10            0              0              A              1
 11            0              0              B              1
 12            0              0              C              1
 13            0              0              D              1
 14            0              0              E              1
 15            0              0              F              1
 16            0              0              10             1
Hard copy? y
```

(A) 正常IDの並び

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 16
NO.            C              H              R              N
 1             0              0              F              1
 2             0              0              2              1
 3             0              0              7              1
 4             0              0              2              1
 5             0              0              3              1
 6             0              0              4              1
 7             0              0              5              1
 8             0              0              6              1
 9             0              0              7              1
 10            0              0              8              1
 11            0              0              9              1
 12            0              0              A              1
 13            0              0              B              1
 14            0              0              C              1
 15            0              0              D              1
 16            0              0              E              1
Hard copy? y
```

(B) IDの絶対位置に関するプロテクトをかけたIDの並び

〔図3-2〕

(2)8876の場合

8876の場合，765のリードダイアグノスティックに相当するのが，リード･トラックコマンドである。

しかし，リード･トラックコマンドは，フォーマットに依存せず，インデックスホール直後から，1トラック分すべてのデータを読み取る。

このため，インデックスホール後，最初のセクターのIDを知るには，読み取ったデータ中から，最初のIDを捜さなければならない。

IBMシステム34フォーマットの場合，読み取ったデータの先頭から，163バイト目がC，164バイト目がH，以後，R，Nとなっている。

しかし，GAP4aの長さが，読み取った場合，必ずしも80バイトあるとは限らないので，実際は，直前のIDAMがどの位置にあるかをサーチして，C，H，R，Nのデータを読み取れば良い。

図3-3に，リード・トラックコマンドで，読み出したデータの先頭部分を示す。

この図では，読み取ったGAP4aの部分の先頭に，1周前のGAP4bが混入しており(6バイト分)，本来80バイトであるはずが，86バイトとなっている。

よって，169バイト目からIDがあり，C＝27H，H＝1，R＝1，N＝1となっている。

この方法により，最初のセクターを知った後，リードアドレスコマンドにて，1トラック分のIDを読み出し，プロテクトをかけた時のIDと一致するかをチェックすれば良い。

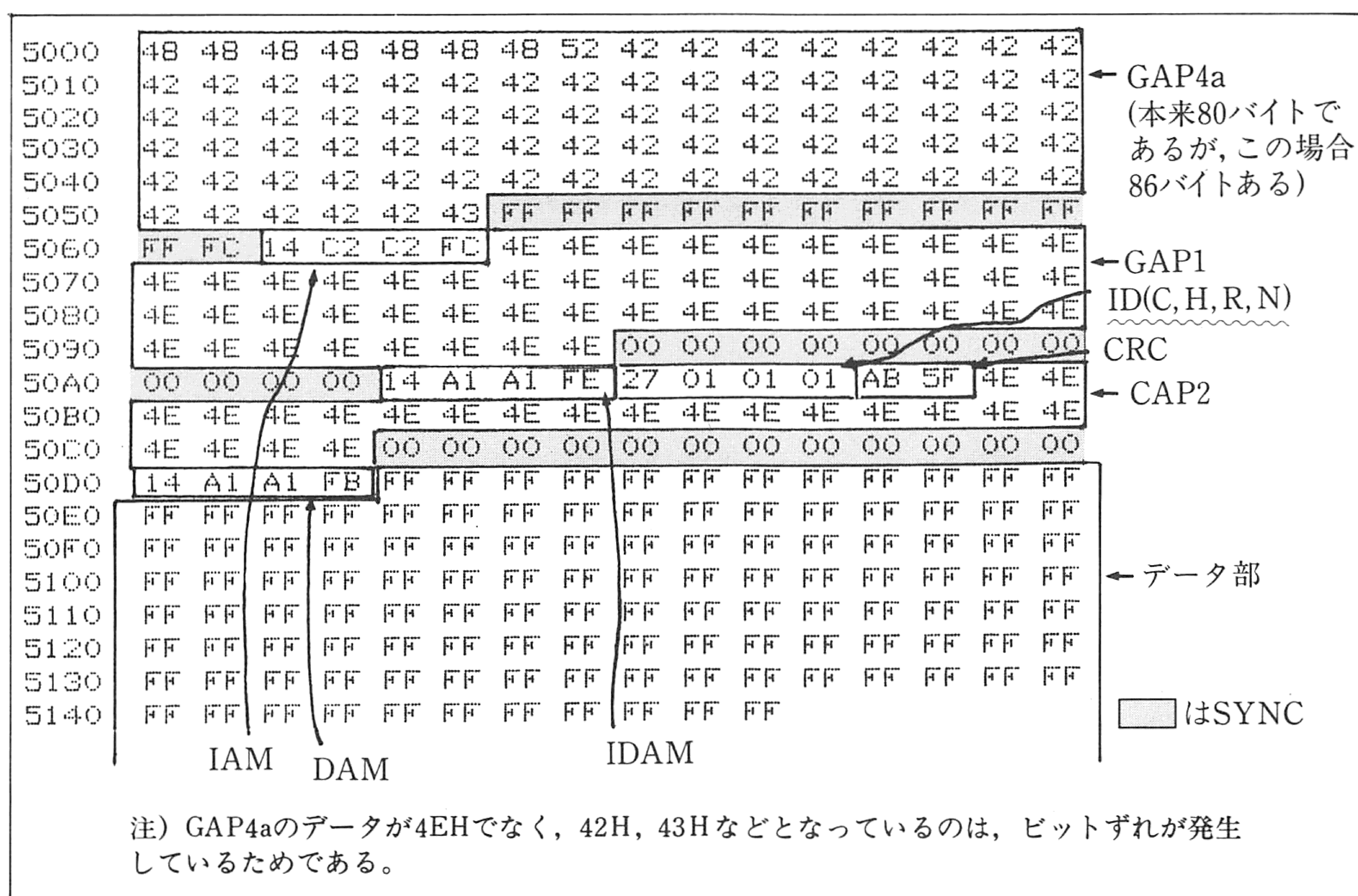
〔補足〕

以上の方法のように，リードダイアグノスティック，リード・トラックを使用せず，リードID，リードアドレスのみによって，インデックス･ホール後最初のセクター（のID）を知る方法を紹介する。

1トラック内の各ブロックの並びは，第1章図1-11-a，bのようになっているが，トラック内最後のセクターと，2周目の最初のセクターの間には，ポストアンブル部，及び，プリアンブル部は存在する（ECMAフォーマットでは，プリアンプル部は存在しない）。

よって，リードIDコマンドないしリード･アドレスコマンドを連続して実行した場合，1番目のセクターのIDを読み出してから次のセクターのIDを読み出されるまでの時間よりも，最後のセクターのIDを読み出してから，2周目の最初のセクターのIDが読み出される時間のほうが長くなるはずである。

また，1番目から2番目のセクターのIDを読み出す時間，2番目から3番目のセクターのIDを読み出す時間………最後から2番目から最後のセクターのIDを読み出す時間は，一定



〔図3-3〕

のはずであり，最後のセクターから，2周目の最初のセクターのIDを読み出す時間のみが大きくなっているはずである。

この事実から，各セクターのIDを読み出すと同時に，読み出しに要した時間を測定することにより，インデック・ホール後，最初のセクターのIDを知ることができる。

3-3　GAP3の長さをチェックするプロテクト

■765，8876

IBM フォーマット，ECMA フォーマットでは，データ長により GAP3 の長さが変化するが，この長さは極端に短い（1バイトぐらいにする）とリード，ライト時に問題となる可能性があるが，長い分には問題がない。

しかし，長くすると，1トラックあたりに入るセクター数が少なくなる。

このプロテクトは，この可変できる GAP 3 の長さをキーワードとして使うもので，今までのプロテクトと異なり，本来データとして使用されていない部分を参照するものである。

(1)765の場合

765の場合，WRITE ID コマンドでフォーマットを行なう際に，GAP3 の長さがパラメータとして指定できるので，これにより GAP3 の長さを決定する。

この場合、全セクターの GAP3 の長さは，同じとなるので，チェックの方法としては，1セクター目の GAP3 の長さをチェックすれば良い。

GAP3 の長さをチェックするには，リードダイアグノスティックコマンドにより，フォーマットされているデータ長よりも大きいNの値で読み出すことにより，まず GAP3 以後までを読み出す。

図 3-4 にN＝1でフォーマットされたトラックに対し，N＝2でリードダイアグノスティックを実行した場合に読み出されたデータを示す。

この後，データ部の CRC の後から，4EH のデータのバイト数が，いくつあるかをカウントすれば良い。

この図のトラックは，フォーマットのみされており，データの書き込みが行なわれていないので，スプライシングポイントによるビットずれが発生していないが，**図 3-5** のように，一度でもセクター対し，データを書き込むと，ビットずれが発生し，GAP3 の値が $4E_H$ ではなくなることが多い。

この場合は，逆に，データ部の CRC の後から，次のセクターの先頭の SYNC までのバイト数をカウントする。

```
4000 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4020 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4030 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4040 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4060 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ←1番目のセクターのデータ
4070 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   GAP3(この場合48バイト)
4100 FB E5 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E    ◣NNNNNNNNNNNNNN
4110 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   NNNNNNNNNNNNNNNN
4120 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   NNNNNNNNNNNNNNNN
4130 4E 4E 00 00 00 00 00 00 00 00 00 00 00 00 A1 A1   NN            ｡｡
4140 A1 FE 00 00 02 01 AF 5F 4E 4E 4E 4E 4E 4E 4E 4E   ｡      ｯ_NNNNNNNN
4150 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 00 00   NNNNNNNNNNNNNN
4160 00 00 00 00 00 00 00 00 00 00 A1 A1 A1 FB FF FF             ｡｡｡
4170 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4180 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4190 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ←2番目のセクター
41A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
     CRC
```

〔図3-4〕

SYNCはビットずれが発生した場合，FF_Hとなるので，これにより認識が可能である。

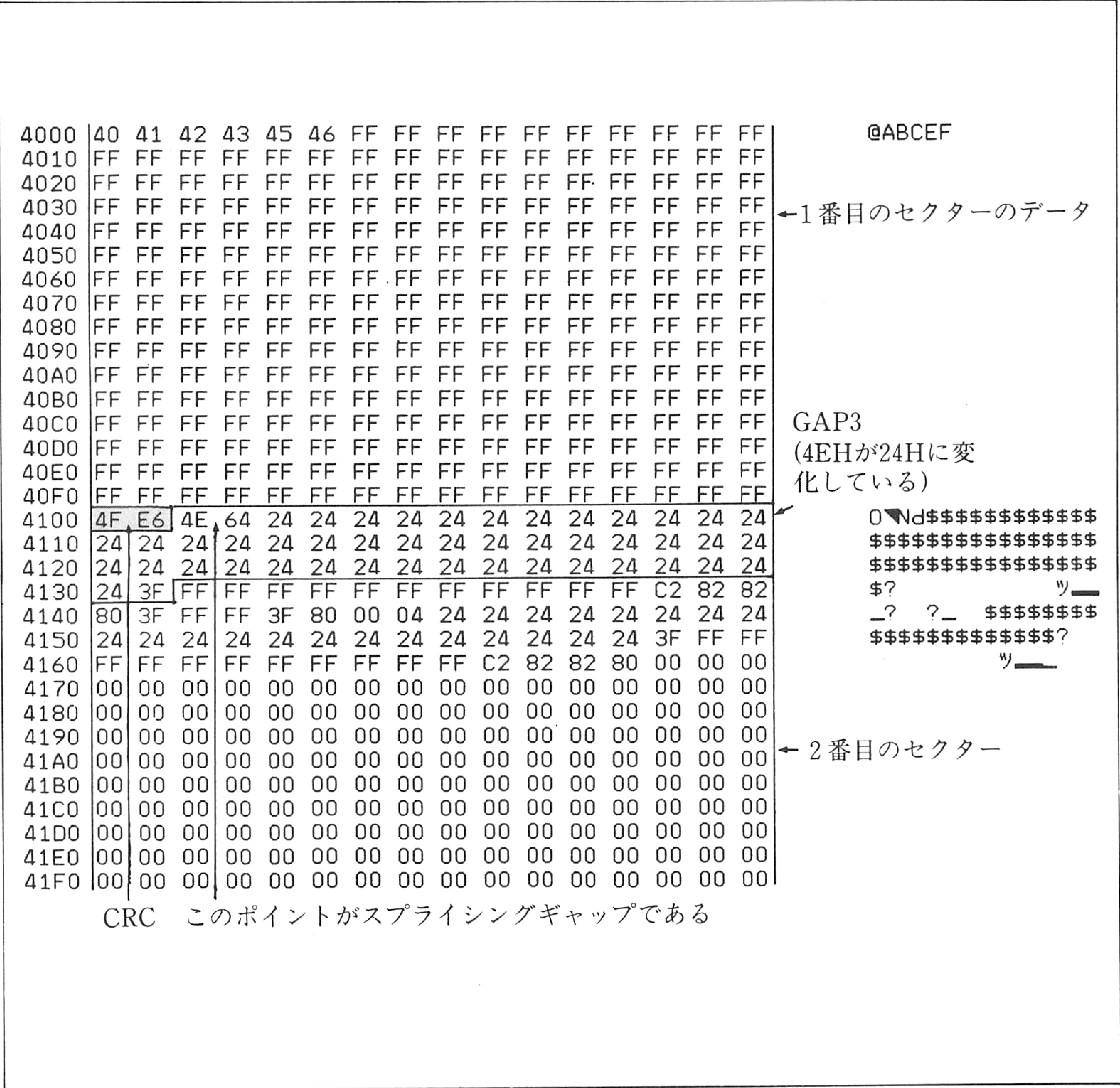
(2)8876の場合

8876の場合は，フォーマット時に，1トラック分のデータをすべてホストから送るので，GAP3の長さもセクターごとに可変させることが可能である。

よって，全セクターのGAP3の長さを変えてフォーマットを行ない，全セクターのGAP3の長さをチェックすれば，より強力なプロテクトとなる。

この方法でフォーマットしたトラックは，765でも読むことが可能で，かつ全セクターのGAP3の長さのチェックも可能である。

しかし，765では，このようなフォーマットは不可能であるので，8876でこのフォーマットをかけたディスケットを765のプロテクトに使用すれば，まず，同一ホストマシンのユーザー



```
4000 40 41 42 43 45 46 FF FF FF FF FF FF FF FF FF FF   @ABCEF
4010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4020 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4030 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4040 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4060 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4070 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4100 4F E6 4E 64 24 24 24 24 24 24 24 24 24 24 24 24   O◥Nd$$$$$$$$$$$$
4110 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24   $$$$$$$$$$$$$$$$
4120 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24   $$$$$$$$$$$$$$$$
4130 24 3F FF FF FF FF FF FF FF FF FF FF FF C2 82 82   $?           ﾂ▬
4140 80 3F FF FF 3F 80 00 04 24 24 24 24 24 24 24 24   _?  ?_  $$$$$$$$
4150 24 24 24 24 24 24 24 24 24 24 24 24 24 3F FF FF   $$$$$$$$$$$$$?
4160 FF FF FF FF FF FF FF FF FF C2 82 82 80 00 00 00            ﾂ▬▬
4170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```


〔図3-5〕

がプロテクトをはずすことは不可能となる。

さて，チェック法であるが，手順としては，765と同じで，リード・トラックコマンドにより，1トラック分のデータを読み出し，データ部のCRCの後から，次のセクターのSYNCまでのバイト数をカウントすれば良い。

図3-6に，リード・トラックで読み出したGAP 3 を示す。

8876の場合，GAP4a の長さが不定であるので，GAP3 の位置をサーチするプログラムに工夫が必要である。

5000	72	72	72	72	72	72	72	77	39	39	39	39	39	39	39	39	
5010	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	
5020	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	
5030	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	
5040	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	→ プリアンブル部
5050	39	39	39	39	39	39	38	00	00	00	00	00	00	00	00	00	
5060	00	00	03	C2	C2	C2	FC	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5070	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5080	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5090	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	
50A0	00	00	00	00	00	14	A1	A1	FE	00	00	01	01	FA	0C	4E	→ ID部
50B0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
50C0	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	00	
50D0	00	14	A1	A1	FB	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
50E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
50F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5100	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5120	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5140	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	→ データ部
5150	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5160	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5170	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5180	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5190	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
51A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
51B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
51C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
51D0	FF	FF	FF	FF	FF	FB	E5	4E	4E	4E	4E	4E	4E	4E	4E	4E	
51E0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	→ GAP3(51バイト)
51F0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5200	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	
5210	00	00	00	00	00	00	14	A1	A1	FE	00	00	02	01	AF	5F	
5220	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5230	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	
5240	00	00	14	A1	A1	FB	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	→ 2番目のセクター
5250	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5260	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5270	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5280	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	

CRC

〔図3-6〕

3-4 1トラックの全セクターのIDを同じにすることによるプロテクト

■765, 8876

このプロテクトは，1トラック内の全セクターのID（C，H，R，N）を同じにし，かつ，目的とするセクター，つまり，インデックス・ホール後，何番目のセクターであるかの指定をして，リード，ライトを行なうものである。

通常のリードデータ，ライトデータコマンドでセクターに対し，リード，ライトが行なわれる場合，まず，ホストから送られたIDとトラック上のIDを比較し，一致したセクターが発見されれば，リード，ライトが行なわれるようになっているが，全セクターのIDが同一の

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 16
NO.            C             H             R             N
 1             0             0             1             1
 2             0             0             1             1
 3             0             0             1             1
 4             0             0             1             1
 5             0             0             1             1
 6             0             0             1             1
 7             0             0             1             1
 8             0             0             1             1
 9             0             0             1             1
 10            0             0             1             1
 11            0             0             1             1
 12            0             0             1             1
 13            0             0             1             1
 14            0             0             1             1
 15            0             0             1             1
 16            0             0             1             1
Hard copy? y
```

〔図3-7〕

場合は，最初に見つけられたセクターに対してリード，ライトを行なわれるため，２度目にリード，ライトする場合は，異なるセクターに対して行なわれる場合がある。

つまり、各セクターのIDが同一のため，各セクターのトラック上の絶対位置がわからないためである。

図 3-7 に同一IDで16のセクターをフォーマットした場合のIDのダンプリスト例，及び**図 3-8** に同一IDのあるセクターに対し，その後，リードを行った場合，絶対位置の異なるセクターの内容が読み出された例を示す。

これらのセクターに対して，インデックス・ホール後X番目の位置にあるセクターに対し，リード，ライトする方法を示す。

このデータを，Ｃ＝０，Ｈ＝０　Ｒ＝１，Ｎ＝１のセクターに書き込む

```
C000 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C010 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C020 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C030 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C040 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C050 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C060 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C070 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C080 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C090 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C0A0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C0B0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C0C0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C0D0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C0E0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
C0F0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUU
```

Ｃ＝０，Ｈ＝０　Ｒ＝１，Ｎ＝１のセクターの内容を読み出すと，同一のセクターのはずであるが違うセクターの内容が読み出されてしまう

```
C000 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C020 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C030 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C040 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C060 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C070 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C0A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C0B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C0C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C0D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C0E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C0F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

〔図3-8〕

(1)765の場合

765の場合の手順を以下に示す。

① リードダイアグノスティックを実行する。

この場合のNは，トラックにフォーマットされているNと等しいか，それより小さい値を使用する。

② ①の直後，リードIDにて（x－2）個のIDを読み出す。ただし，2番目のセクターの場合，IDは読まず，次のステップへ進む。

③ ②の直後，ライトデータ，またはリードデータを実行する。

各手順は，次のような意味である。

①では，リードダイアグノスティックにより，インデックスホール後最初のセクターをサーチする。

②では，①の直後（この点が重要である）（x－2）個のIDを読み飛ばす。

たとえば，5番目のセクターをアクセスする場合は，3個のIDを読み飛ばすので，次のセクターは，通算，目的の5番目のセクターとなる。

③では，②の手順にて，次のセクターが目的のセクターであるように「位置決め」がされているので，②の手順の直後，リード，ライトを行なえば，目的のトラックに対して，リードが可能である。

また，1番目から，最後のセクターまでのデータをすべて読み出すのであれば，①～③の手順を繰り返さず，③の手順に，マルチセクターリードを使用すれば良い。

データを書き込む場合は，同様にマルチセクターライトを使用する。

(2)8876の場合

8876の場合の手順を以下に示す。

① 「IDの絶対位置に関するプロテクト」の補足の方法により，インデックスホール後最初のセクターのサーチを行なう。

② ①の直後，リードアドレスにより，（x－2）個のIDを読み出す。ただし，x＝2の場合，IDは読まずに，③のステップに進む。

③　②の直後，ライトデータ，またはリードデータを実行する。

765の手順と異なるのは，①で，これは，8876がインデックス・ホール後最初のセクターに対し，何らかの処理を行なうコマンドを持たないためである。

前述の補足の方法では，IDの読み出し時間の比較により最初のセクターをサーチしたが，あらかじめ最初のセクターのID読み出し時間と，他のセクターの読み出し時間を確認しておき，ある時間以上かかって読み出されたIDのセクターは，最初のセクターであると認識する方法も考えられ，むしろ実用的であろう。

このプロテクトの変形として，全セクターのIDを同一とせず，いくつかのセクターのIDを同一とする方法もある。

この場合は，同一IDのセクターの1つ前のセクターは，すべて異なるIDを持たせておけば，リード，ライトは，まずダミーで，1つ前のセクターをリードし，その直後に，同一IDのセクターにリード，ライトすることにより可能となり，手順も簡単である。

3-5　1トラック内に異なるデータ長のセクターを使用するプロテクト①

■765，8876

通常，同一トラック内のセクターのデータ長はすべて同一となっているが，各セクターごとにデータ長を変えることによりかけるプロテクトである。

原理的には，フォーマット時に各セクターのデータ長が可変できれば，問題なくリード，ライトが可能である。

しかし，このフォーマッティングが可能なのは，8876など，フォーマットのデータをすべてホストマシンから送れるものでなければ不可能で，765のように，マクロ的なパラメータを送ることのみで，フォーマットを行なうFDCでは違った方法が必要である。

以下に765を使用した場合の方法を解説する。

例として，**図3-9**に示すIDを持つ8個のセクターが正常にリード，ライトが可能になるようにフォーマットする場合を考える。

この例では，2番目のセクターのデータ長がN＝2となっており，他はN＝1となっている。

まず，図3-9に示すIDを使用して，N＝1でフォーマットを行ない，次に，2番目のセ

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 8
NO.           C          H          R          N
 1            0          0          1          1
 2            0          0          2          2   ← このセクターのみ N＝2
 3            0          0          3          1
 4            0          0          4          1
 5            0          0          5          1
 6            0          0          6          1
 7            0          0          7          1
 8            0          0          8          1
Hard copy? y
```

〔図3-9〕

クターに対し，ライト動作を行なう。

この後，再びIDを読み出すと，**図3-10**のように，N＝2のセクターの次のセクターが消えてしまっている。

これは，N＝1でフォーマットされているセクターに対して，N＝2でライト動作を行なったため，次のセクターのID部のエリアにまでデータが書き込まれてしまい，見かけ上，次のセクターが消えてしまったと認識されるためである。

これを避けるためには，**図3-11**のように，N＝2のセクターの次にダミーとなるセクターを設けて，フォーマットを行ない，その後，N＝2のセクターにライト動作を行なえば，ダ

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 7
NO.            C            H            R            N
 1             0            0            1            1
 2             0            0            2            2
 3             0            0            4            1 ← R＝3のセクターが
 4             0            0            5            1   消えてしまっている
 5             0            0            6            1
 6             0            0            7            1
 7             0            0            8            1
Hard copy? y
```

〔図3-10〕

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 9
NO.            C            H            R            N
 1             0            0            1            1
 2             0            0            2            2
 3             0            0            2            0 ← N＝2のセクター
 4             0            0            3            1   の後にダミーのセ
 5             0            0            4            1   クターを追加する。
 6             0            0            5            1   C，H，R，Nの
 7             0            0            6            1   値は他のセクター
 8             0            0            7            1   と違っていれば何
 9             0            0            8            1   でも良い。
Hard copy? y
```

〔図3-11〕

ミーのセクターが消えるが，目的とするフォーマットが得られる(図 3-12)。

この手法を用いて，データ長の異なるセクターを1トラック内に作成することができるが，いくつかの注意点を示しておく。

まず，アンフォーマット時の1トラックの容量は，6250 (186H) バイトとなっているので，これを越えないことである（5インチ，MFM時)。

また，データ長を可変してライト動作を行なった場合，データ長によっては上記の例のように次のセクターだけでなく，数セクターに渡り消される場合があるので，あらかじめ，何セクター分が消されるか計算するか，実験にて確認を行なうこと。

また，GAP3の長さも考慮に入れなければならない。

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 8
NO.          C          H          R          N
 1           0          0          1          1
 2           0          0          2          2  ← ダミーのセクター
 3           0          0          3          1    が消えて，目的の
 4           0          0          4          1    フォーマットが得
 5           0          0          5          1    られる。
 6           0          0          6          1
 7           0          0          7          1
 8           0          0          8          1
Hard copy? y
```

〔図3-12〕

3-6　1トラック内に異なるデータ長のセクターを使用するプロテクト②

■765

前出の**図 3-12**のフォーマットを行なう場合，ダミーのセクターを設けて，N＝1でフォーマットする方法を解説したが，同じフォーマットを行なうのに，もう一つ別の方法がある。

前出の方法は，1トラックのセクター中，データ長の一番短い値（例ではN＝1）にてフォーマットを行ない，ライト動作により，ダミーのセクターを消去してフォーマットを完成させる，という方法であったが，逆にセクター中，データ長の一番長い値（例ではN＝2）にてフォーマットを行なう方法である。

この場合，N＝2のセクターは，そのデータ長でフォーマットがされているため，リード，ライト動作は正常に行なわれるが，N＝1のセクターは，一度データを書き込んでやらないと，問題が発生する。

しかし，この2つの方法で1トラック内にデータ長のセクターを存在させるフォーマットを行った場合，両方とも，すべてのセクターに対して正常にリード，ライトが行なわれるので，見分けが付かない。

そこで，さらにこの2つの手順の違いをキーワードとするプロテクトを考える。

つまり，セクターのデータ長を変化させることで，まず，1次のプロテクトを行ない，さらにフォーマットの作成手順の違いをチェックする2次のプロテクトをかけるわけである。

例として，**図 3-9** のフォーマットで考える。

まず、このフォーマットを上記の2つの手順により作成する。

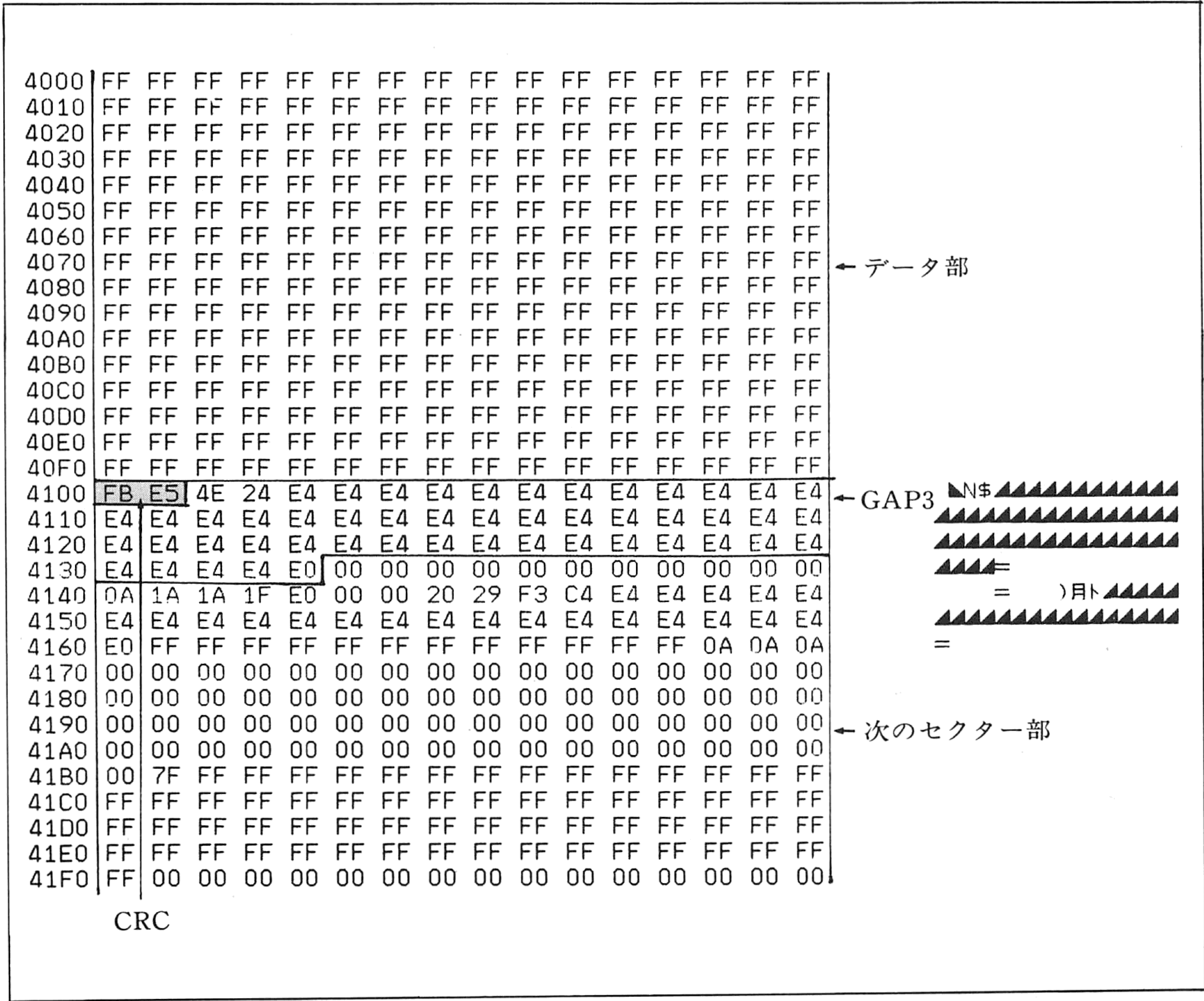
そして，リードダイアグノスティックコマンドにより各512バイト分のデータを読み出したものが，**図 3-13**及び**図 3-14**である。

図 3-13 は，N＝1でフォーマットした手順のもので各データは，何ら異常な点はないが，**図 3-14** のN＝2でフォーマットした手順のものは，データに異常が見られる。

つまり，データ部のCRCの後，GAP3がなく，N＝2でフォーマットしたときの，254バイト＋2バイトのCRCが残ってしまっているのである。

これは，リード，ライト時には，何ら問題が発生しないので，このような方法により確認しなければならない。

逆に，この方法のように，リードダイアグノスティックにより，データ部のCRCの後にGAP3が存在するかどうかにより，2つの手順のいずれかを使用したかのチェックを行なうことが可能である。



```
4000 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4020 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4030 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4040 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4060 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4070 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4100 FB E5 4E 24 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4
4110 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4
4120 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4
4130 E4 E4 E4 E4 E0 00 00 00 00 00 00 00 00 00 00 00
4140 0A 1A 1A 1F E0 00 00 20 29 F3 C4 E4 E4 E4 E4 E4
4150 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4 E4
4160 E0 FF FF FF FF FF FF FF FF FF FF FF FF 0A 0A 0A
4170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41B0 00 7F FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41F0 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```


〔図3-13〕

4000	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4100	FB	E5	4E	00	00	00	00	00	00	00	00	00	00	00	00	00
4110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	01

← データ部

N

← N＝2でフォーマットしたときのデータ部の残り
512－(256＋2)
＝254バイト

CRC

〔図3-14〕

3-7　データ部のCRCエラーを利用するプロテクト

■765，8876

データ部，IDの部のCRCエラーは，主にメディアの不良により発生するものであるが，これを人為的に作り，プロテクトに利用する方法である。

データ部のCRCは，ライト動作(ライトデータコマンドにより，セクターのデータ部に書き込みを行なう）時に自動的にFDC内で計算され，書き込まれる。

このため，メディアが不良でなければ，データ長にかかわらず，一担ライト動作が行なわれるとデータ部のCRCエラーは発生しなくなる。

よってデータ部のCRCエラーを作成するには，フォーマット時に作成する方法が一般的である。

765の場合は，ID部で指定するデータ長と異なるデータ長でフォーマットを行なえば，フォーマットのみしただけの状態でリード動作を行えば，データ部CRCエラーが発生する。

これは当然のことで，データ長が，例えば，N＝1でフォーマットされているセクターに対し，ID部がN＝2であれば，リードされるデータ長は，512バイトが読み取られるので，FDCは，本来CRCでない部分をCRCとしてチェックを行なうため，エラーが発生するわけである。

8876の場合は，フォーマット時にCRCを書き込むデータ（$F7_H$）を送らず，代りに，ダミーのデータを送ればデータ部のCRCエラーを作成できる。

765と異なるのは，データ長がID部と同じであっても，CRCエラーが作成できる点である。

ところで，データ部のCRCエラーは，データが読み取られた後に発見されるため，同時にデータの内容に対してもチェックを行ないプロテクトをかけることが可能である。

765の場合は，IDのデータ長が，実際にフォーマットされているデータ長に対して，大きいか小さいかによって，読み取られるデータが異なるし，8876の場合，フォーマット時に同時にデータ部のデータを書き込め（ただし，コントロールデータの値は書き込めない）ば良い。

8876でこのようにしてかけたプロテクトは，765でチェックすることは可能であっても，同じフォーマットで書き込むことは不可能であるので，強力なプロテクトとなる。

3-8 ID部のCRCエラーを利用するプロテクト

■8876

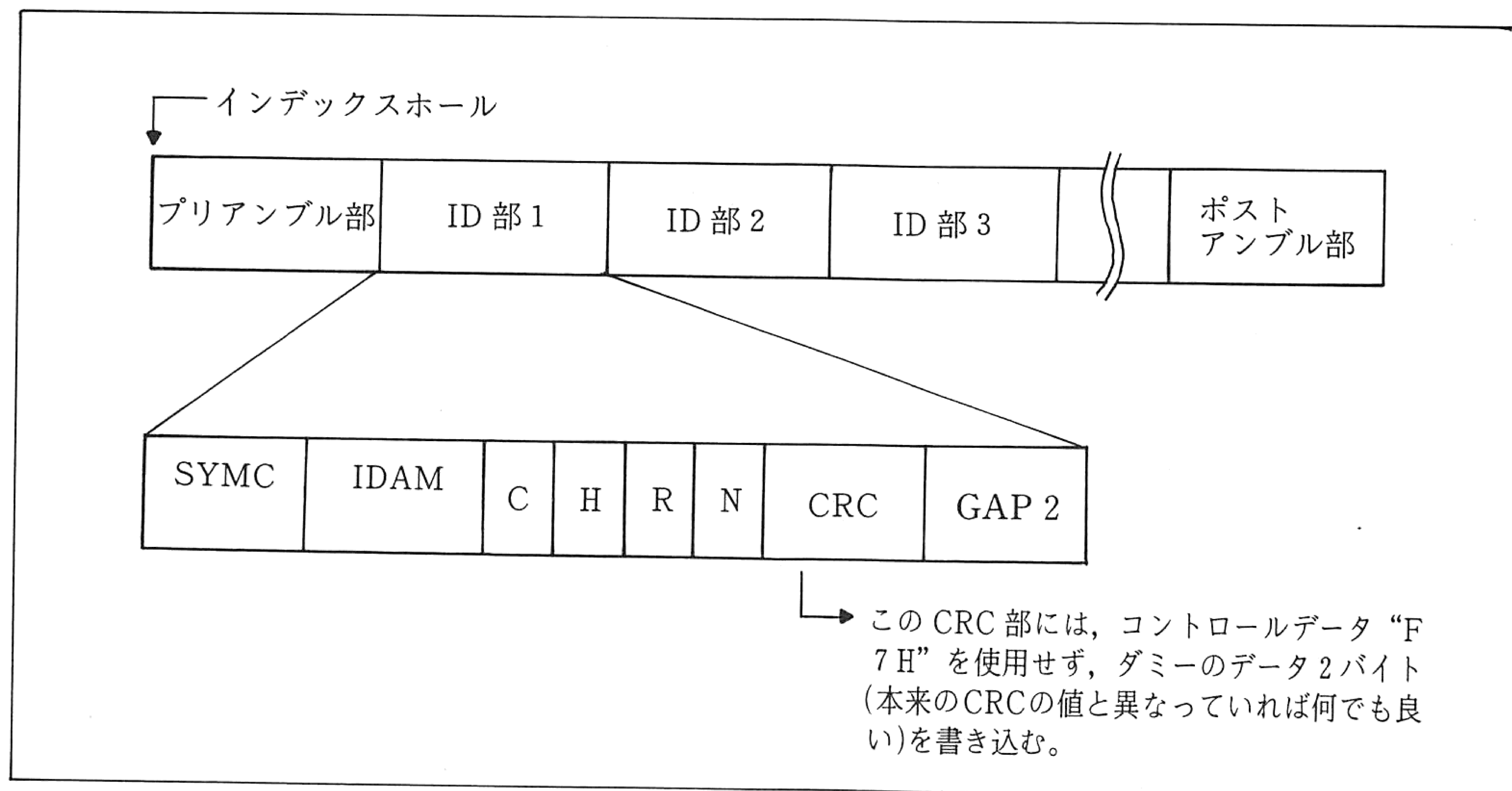
8876は，データ部のCRCエラーを作成する方法と同様の方法でID部のエラーを作成することが可能である。

リード動作では，ID部にCRCエラーがあると，続くデータ部は読み取られず，ID部のCRCエラーが発見された時点で，コマンドが異常終了されてしまう。

よって，この場合，リード・トラックコマンドでデータ部を読まないかぎり，データ部があろうとなかろうと同じである。

よって，**図3-15**のようなフォーマットを使用すればより多くのID部が入るので，これにより，2次的なプロテクトをかけることも可能である。

このプロテクトは，765でもチェックすることは可能であるが，フォーマットを作成することは不可能であるので，765ではこのプロテクトは，はずせないことになる。



〔図3-15〕

3-9 データ部が読み取れないことを利用するプロテクト

■765，8876

リード動作を行なった場合，データ部が読み取れないことを利用するプロテクトである。

データ部が読み取れない（FDCのリザルトステータスの該当するエラービットが立つ）原因としては，DAMまたは，DDAMが正常に読み取れなかったか，DAM，またはDDAMが存在しないかのいづれかである。

よって，これと同じ状態を人為的に作り出せば良いわけであるが，この手順は，FDCによって異なるので，個別に解説する。

(1)765の場合

765の場合，WRITE IDコマンドによりフォーマッティングを行なうが，このコマンドは，マクロパラメータをFDCに送るだけでフォーマッティングを行なうものであるので，DAM，ないしはDDAMを存在させないようにフォーマッティングすることは不可能である。

また，ライト動作時も，メディアが不良でないかぎり，正常にDAMまたはDDAMが書き込まれてしまうので，一見DAM，DDAMを操作することは不可能と思われるが，実は実験を行なうと，面白い事実がわかった。

765の場合，N＝0でフォーマットを行なうとデータ長128バイト／セクターでフォーマットが行なわれる。

これは，MFM，FMモードいづれも同じであり，正常にフォーマットが行なわれる。

しかし，N＝0の場合，MFMモードにて，ライト動作を行なうと，DAM，DDAMが破壊され，この後，リード動作を行なうと，データが読み出せないこととなった。

よって，N＝0と限定はされるが，765でもこのプロテクトがかけられることとなる。

図3-16に，この方法によって，DAMを読み出し不能としたセクターのダンプを示す。

765のリードダイアグノスティックでは読み出せないので，8876のリード・トラックによって読み出している。

8876のリード・トラックコマンドで読み出されるデータは，ミッシングクロックのデータ，つまり，アドレスマークに同期して読み出されるが，図のDAM部分はビットずれが発生し

ており，これではFDCがDAMとして認識できないことがわかる。

これは筆者の想像であるが，おそらく，765はN＝0の場合，MFMモードでライト動作を行なうと，DAMのA1H3バイトを書き込んだ時点でライト動作が異常終了するためではないかと思われる。

しかし，765のマニュアルには，MFMモード時にN＝0のデータ長を使用することについては規定されておらず，一概にLSIのロジックミスと考えるのは不適当である。

(2)8876の場合

8876の場合は，自由のフォーマットでフォーマッティングすることが可能であるので，DAM, DDAMを書かないことも，またDAM, DDAMを765の場合と同じように不完全に書き込む（DAMのFB$_H$ないしはF8$_H$のかわりにダミーの値を書き込めば良い）ことも可能である。

特に，DAMを書かなければデータ部は必要ないので，図 3-15のフォーマットを使用することもできる（ただし，ID部のCRCはコントロールデータ“F7$_H$”を使用して正常に書かれるようにする）。

逆に，データ部を残しておき，リード・トラック・コマンドにて，読み出し，チェックする，といった2次のプロテクトを使用することも可能である。

```
5000  90 90 90 90 90 90 90 90 8E 4E 4E 4E 4E 4E 4E 4E
5010  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
5020  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
5030  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   ← プリアンブル部
5040  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
5050  4E 4E 4E 4E 4E 4E 4E 00 00 00 00 00 00 00 00 00
5060  00 00 00 C2 C2 C2 FC 4E 4E 4E 4E 4E 4E 4E 4E 4E
5070  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
5080  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
5090  4E 4E 4E 4E 4E 4E 4E 4E 4E 00 00 00 00 00 00 00   IAM
50A0  00 00 00 00 00 14 A1 A1 FE 00 00 01 00 EA 2D 4E   正常に書かれている
50B0  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
50C0  4E 4E 4E 4E 4E 00 00 00 00 00 00 00 00 00 00 00
50D0  00 05 A1 A1 FD F7 77 77 F7 77 77 77 77 77 77 77
50E0  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77
50F0  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77
5100  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77
5110  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77
5120  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77
5130  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77
5140  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77
```

破壊されたDAM。
本来ならば，FDHは，FBHでなければならない。
かつ，この時点でビットずれが発生している。
(この時点でコマンドが異常終了しているとも考えられる)

〔図3-16〕

3-10 スプライシング・キャップ(ビットずれ)を利用したプロテクト

■765, 8876

スプライシング・ギャップという言葉は，著者の造語で，あるセクターにデータが書き込

```
4000 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4020 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4030 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4040 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4060 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4070 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4100 FB E5 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   ◣NNNNNNNNNNNNNN
4110 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   NNNNNNNNNNNNNNNN
4120 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   NNNNNNNNNNNNNNNN
4130 4E 4E 4E 4E 4E 00 00 00 00 00 00 00 00 00 00 00   NNNNN
4140 00 A1 A1 A1 FE 00 00 02 01 AF 5F 4E 4E 4E 4E 4E    ｡｡｡        ｯ_NNNNN
4150 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   NNNNNNNNNNNNNNNN
4160 4E 00 00 00 00 00 00 00 00 00 00 00 00 A1 A1 A1   N             ｡｡｡
4170 FB FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4180 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4190 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

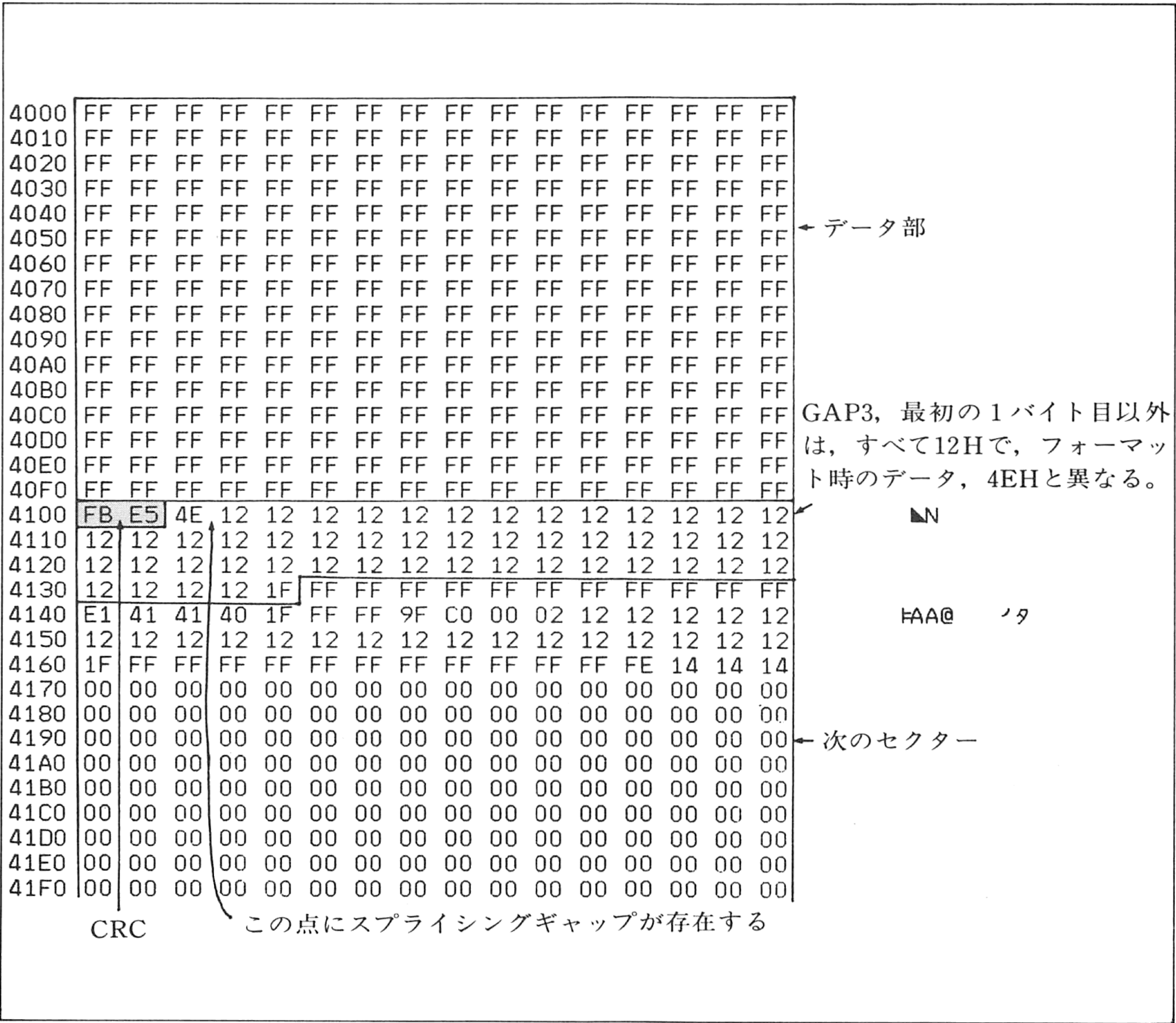

〔図3-17〕スプライシング・ギャップのない場合(μPD-765A)

まれた場合，その前後にて発生するギャップ（ずれ）をこう称している。

このプロテクトは，トラック上に，このスプライシング・ギャップが存在するか否かをチェックするプロテクトである。

スプライシング・ギャップは，フォーマットのみを行なった（ライト動作を行なっていない）トラックには存在せず，かつ，ライト動作を1度も行なっていないセクターの前後にも存在しない。

チェック方法としては，765の場合はリードダイアグノスティックコマンド，8876の場合は，リード・トラックコマンドにより，GAP3 の値が，フォーマット時の値(MFM なら $4E_H$)と同じかどうかをチェックする。



```
4000 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4020 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4030 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4040 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4060 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4070 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4100 FB E5 4E 12 12 12 12 12 12 12 12 12 12 12 12 12
4110 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
4120 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
4130 12 12 12 12 1F FF FF FF FF FF FF FF FF FF FF FF
4140 E1 41 41 40 1F FF FF 9F C0 00 02 12 12 12 12 12
4150 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
4160 1F FF FF FF FF FF FF FF FF FF FF FF FE 14 14 14
4170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```


〔図3-17〕スプライシング・ギャップのある場合(μPD-765A)

同じであれば，スプライシング・ギャップは存在していないと判断する。

ところで，8876のリード・トラックコマンドでは，アドレスマークが発見されると，それに同期が取られるので，1セクター目にスプライシング・ギャップがあっても，2セクター目のDAMで同期が取られるため，2セクター目のGAP3をチェックして，2セクター目のスプライシング・ギャップの存在がチェックできるが，765のリードダイアグノスティックコマンドでは，データの読み出しの先頭，つまり，インデックスホール後，最初のセクターの

5000	4E	4E	4E	4E	4E	4E	4E	4E	72	72	72	72	72	72	72	72	
5010	72	72	72	72	72	72	72	72	72	72	72	72	72	72	72	72	
5020	72	72	72	72	72	72	72	72	72	72	72	72	72	72	72	72	
5030	72	72	72	72	72	72	72	72	72	72	72	72	72	72	72	72	
5040	72	72	72	72	72	72	72	72	72	72	72	72	72	72	72	72	←プリアンブル部
5050	72	72	72	72	72	72	70	00	00	00	00	00	00	00	00	00	
5060	00	00	06	C2	C2	C2	FC	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5070	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5080	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5090	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	
50A0	00	00	00	00	00	14	A1	A1	FE	00	00	01	01	FA	0C	4E	←ID部
50B0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
50C0	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	00	
50D0	00	14	A1	A1	FB	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
50E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
50F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5100	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5120	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	←データ部
5140	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5150	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5160	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5170	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5180	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5190	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
51A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	CRC
51B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
51C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
51D0	FF	FF	FF	FF	FF	FB	E5	4E	4E	4E	4E	4E	4E	4E	4E	4E	
51E0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	←GAP3，データは4EHでフォーマット時と同じである。
51F0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5200	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	
5210	00	00	00	00	00	00	14	A1	A1	FE	00	00	02	01	AF	5F	
5220	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5230	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	←次のセクター
5240	00	00	14	A1	A1	FB	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
5250	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	

〔図3-18〕スプライシング・ギャップのない場合(MB8876A)

DAM，ないしはDDAMで同期が取られるだけで，以後はアドレスマークがあっても同期は取られない。

このため，あるセクターにスプライシング・ギャップが存在すると，それ以後は，ビットがずれてデータが読み出されてしまうので，それ以後のセクターにスプライシング・ギャップがあろうとなかろうと，GAP3の値はフォーマット時の値と異なってしまう。

5000	21	21	21	21	21	21	21	20	09	09	09	09	09	09	09	09
5010	09	09	09	09	09	09	09	09	09	09	09	09	09	09	09	09
5020	09	09	09	09	09	09	09	09	09	09	09	09	09	09	09	09
5030	09	09	09	09	09	09	09	09	09	09	09	09	09	09	09	09
5040	09	09	09	09	09	09	09	09	09	09	09	09	09	09	09	09
5050	09	09	09	09	09	09	0F	FF	FF	FF	FF	FF	FF	FF	FF	FF
5060	FF	FF	F1	14	C2	C2	FC	4E	4E	4E	4E	4E	4E	4E	4E	4E
5070	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
5080	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
5090	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00
50A0	00	00	00	00	00	14	A1	A1	FE	00	00	01	01	FA	0C	4E
50B0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
50C0	4E	4E	4E	4E	4E	07	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
50D0	FF	C2	A1	A1	FB	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
50E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
50F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5100	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5120	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5140	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5150	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5160	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5170	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5180	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5190	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
51A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
51B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
51C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
51D0	FF	FF	FF	FF	FF	FB	E5	4E	64	24	24	24	24	24	24	24
51E0	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
51F0	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
5200	24	24	24	24	24	24	24	24	24	3F	FF	FF	FF	FF	FF	FF
5210	FF	FF	FF	FF	FF	C2	A1	A1	FE	00	00	02	01	AF	5F	4E
5220	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
5230	4E	4E	4E	4E	4E	03	00	00	00	00	00	00	00	00	00	00
5240	00	14	A1	A1	FB	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5250	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

〔図3-18〕スプライシング・ギャップのある場合(MB8876A)

よって，765を使用する場合，この点に注意してプロテクトをかける必要がある。

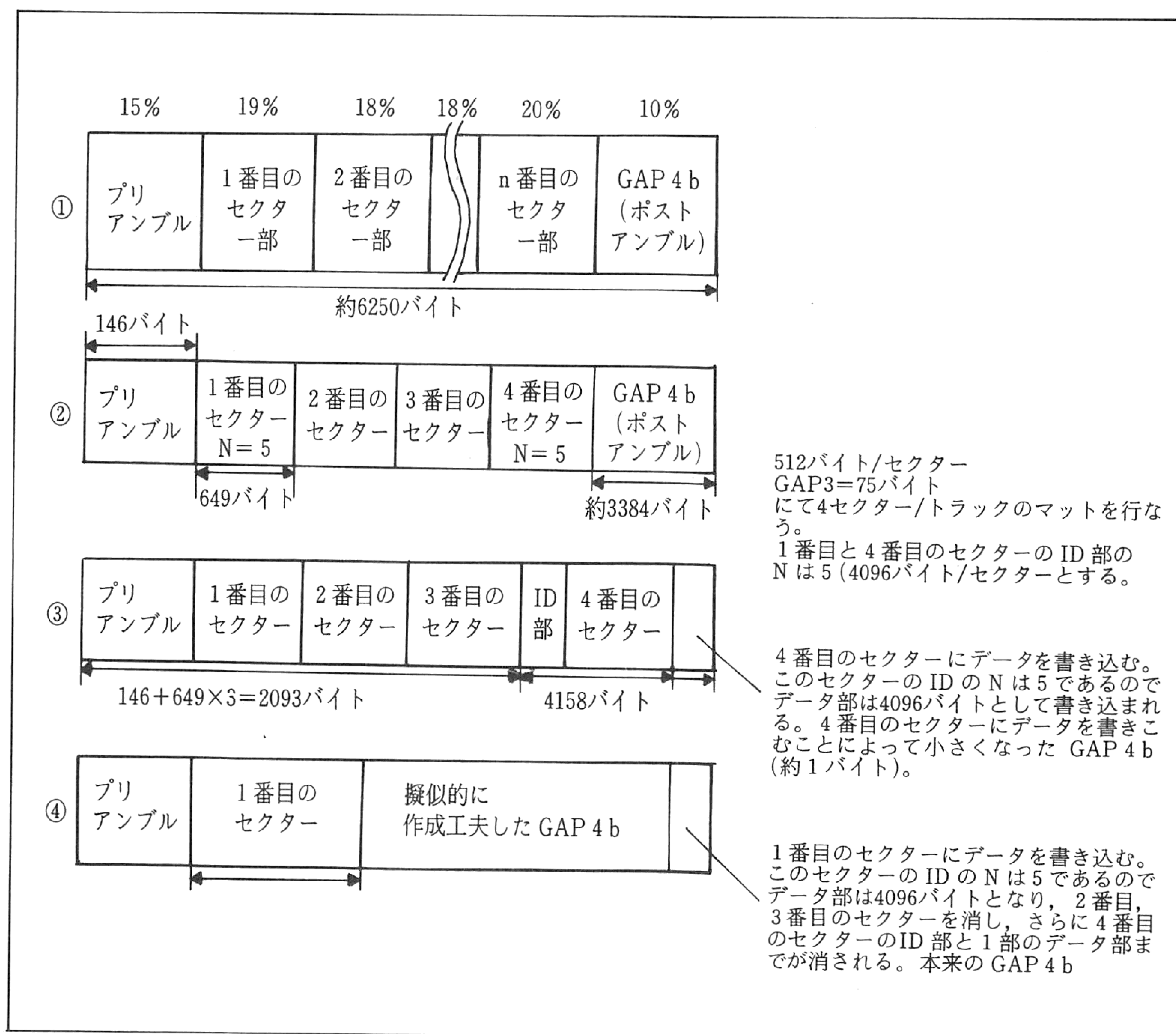
図3-17に765のリード・ダイアグノスティックで読んだ，スプライシング・ギャップのある場合と，ない場合のダンプリスト，**図3-18**に8876のリード・トラックで読んだ場合の同ダンプリストを示す。

このプロテクトは，けっこう強力で，GAP3の値をこの方法で読み出し，チェックしないかぎり，プロテクトがかかっていないように見えるといった利点を持つ。

3-11　GAP4bの値を変えることによるプロテクト

■765，8876

このプロテクトは，通常では使用されないGAP4b（ポストアンブル部）の値を変えること



〔図3-19〕

によってかけるプロテクトである。

GAP4b の値は通常 $4E_H$ であるが，この値に他の値を使用してフォーマットを行なう。

8876の場合は，ライト・トラックにてフォーマットを行なう場合に該当するデータに $4E_H$ を使用せず他の値を使用してフォーマットを行なうだけである。

しかし，765の場合は、WRITE ID にてフォーマットを行なうと，GAP4b の値は自動的に $4E_H$ となってしまうため，フォーマット時に GAP4b の値を変えることは不可能である。よって，765にてこのフォーマットを作成するためには次のような手順を使用する。

図3-19，①のように5インチ，MFM の場合，1トラックあたりの容量は6250バイト＋α となっている。

ここで，4096バイト／セクター，1セクター／トラックの場合に GAP4b の値を変化させる方法を例にして考える。

まず，②のように512バイト／セクター，GAP3＝75バイトにて4セクター／トラックのフォーマットを WRITE ID によって作成する。

この場合1番目と5番目のセクターの ID 部のNを5とする。

他のセクターの ID パラメータは何でも良い。

次に③のように4番目のセクターにデータを書き込む。

ID 部のNの値が5であるため，書き込みは4096バイト分行なわれる。

このため，GAP4b の部分まで書き込みが行なわれるので，GAP4b は約1バイト（実際はもっと多い）と小さくなる。

また，この時に書き込んだデータが擬似 GAP4b となるので，データとしては，00H や FFH（スプライシングギャップによるビットずれが発生した場合でも認識しやすい）などの単一データを使用すると良い。

また単一データ中に，キーワード的に数バイト異なったデータを使用するのも効果的である。

この後，さらに1番目のセクターに対してデータの書き込みを行なう（図の④)。

このセクターの ID 部のNも5であるので，データ部に4096バイト分が書き込まれる。

このため，2番目，3番目のセクター，及び4番目のセクターの ID 部とデータ部の1部が消去される。

これにより，残った4番目のセクターのデータ部が擬似的な GAP4b として作成されたわけである。

次にチェックの方法であるが，これは GAP4b の値を，765の場合は，リードダイアグノスティックで，8876の場合はリードトラックにて読み出し，本来の値であるかどうかのチェックを行えば良い。

ここで注意しなければならないのは，GAP4b は，スプライシングギャップがあると 4EH 以外の値，たとえば 27H などになるので，逆に擬似的に作成した GAP4b の値であるかどうかのチェックを行なう方が簡単である。

擬似的に作成された GAP4b の値として 00H，または FFH が使用されている場合は，ビットずれが発生しても，必ず 00H または FFH となるからである。

このプロテクトは765で使用する場合，手順が複雑であるが，逆に強力である(解析されにくい）のでその効果は大きい。

図 3－20 にこの方法で作成した GAP4b のダンプリストと，通常時の GAP4b のダンプリ

スプライシングギャップがない場合

```
4F90 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|
4FA0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|
4FB0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|
4FC0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|
4FD0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|←データ部
4FE0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|
4FF0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|  GAP4bの値は4EH
5000 |35 35|4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|←   55NNNNNNNNNNNNNN
5010 |4E|4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|    NNNNNNNNNNNNNNNN
5020 |4E|4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|    NNNNNNNNNNNNNNNN
5030 |4E|4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|    NNNNNNNNNNNNNNNN
5040 |4E|4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|    NNNNNNNNNNNNNNNN
5050 |4E|4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|    NNNNNNNNNNNNNNNN
5060 |4E|4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|    NNNNNNNNNNNNNNNN
5070 |4E|4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|    NNNNNNNNNNNNNNNN
     CRC
```

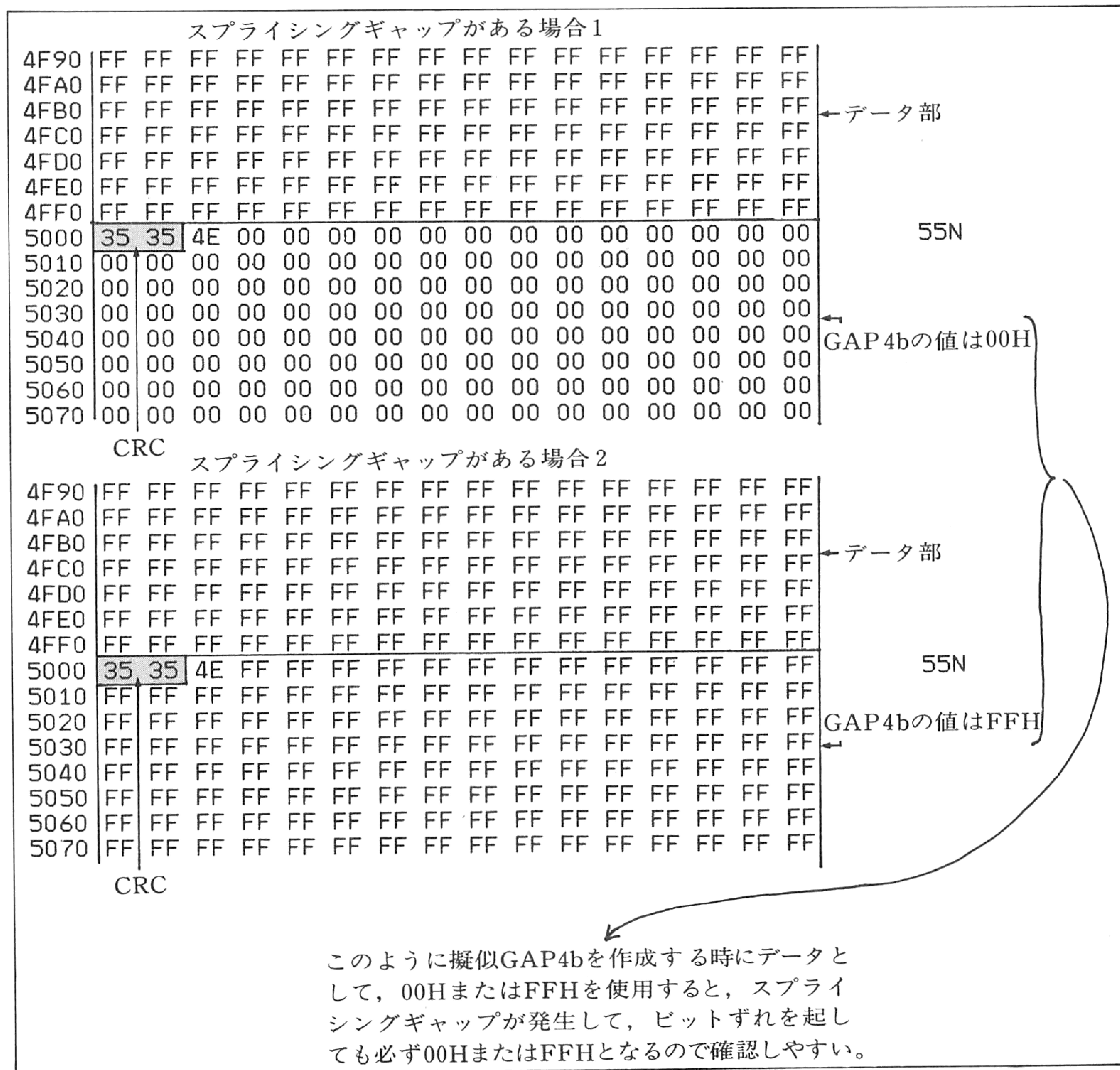
スプライシングギャップがある場合

```
4F90 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|
4FA0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|
4FB0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|
4FC0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|←データ部
4FD0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|
4FE0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|  GAP4bの値は27H
4FF0 |FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF|  (ビットずれが発生)
5000 |35 35|4E 27 27 27 27 27 27 27 27 27 27 27 27 27|←   55N'''''''''''''
5010 |27|27 27 27 27 27 27 27 27 27 27 27 27 27 27 27|    ''''''''''''''''
5020 |27|27 27 27 27 27 27 27 27 27 27 27 27 27 27 27|    ''''''''''''''''
5030 |27|27 27 27 27 27 27 27 27 27 27 27 27 27 27 27|    ''''''''''''''''
5040 |27|27 27 27 27 27 27 27 27 27 27 27 27 27 27 27|    ''''''''''''''''
5050 |27|27 27 27 27 27 27 27 27 27 27 27 27 27 27 27|    ''''''''''''''''
5060 |27|27 27 27 27 27 27 27 27 27 27 27 27 27 27 27|    ''''''''''''''''
5070 |27|27 27 27 27 27 27 27 27 27 27 27 27 27 27 27|    ''''''''''''''''
     CRC
```

〔図3-20〕通常のフォーマットで作成されたGAP4b

ストを示す。

これらは，いずれも765にて作成したものである。



〔図3-20〕図3-19の方法により作成したGAP4b

3-12 一見アンフォーマットに見えるプロテクト

■8876

8876ではIBM，ECMA フォーマット以外のフォーマットが可能である。

また，リード・トラックにより，フォーマットに係わらず1トラック内のすべてのデータを読み出すことが可能である。

そこで，通常のリード，ライト動作で使用される各セクターのID部を作成せずデータを書き込めば，リード・トラックにて読み出さないかぎり，一見アンフォーマットを見られるといったプロテクトを作成することが可能である。

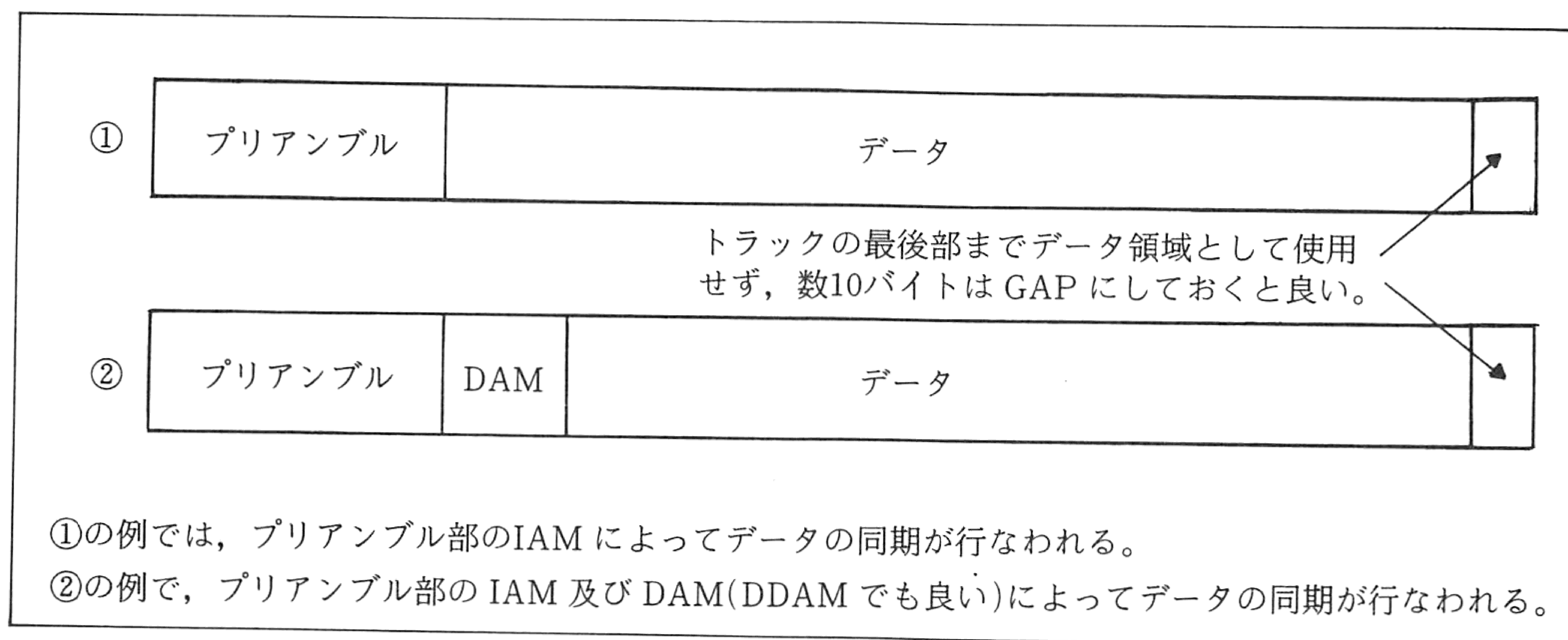
この場合，注意しなくてはならないのは，トラックの最初に少なくとも1つ以上のアドレスマークが存在しないと，リードトラック時にデータの同期が取れず，ビットがずれた状態でデータが読み出されてしまう。

よって，プリアンブルに相当する部分が少なくとも必要になる。

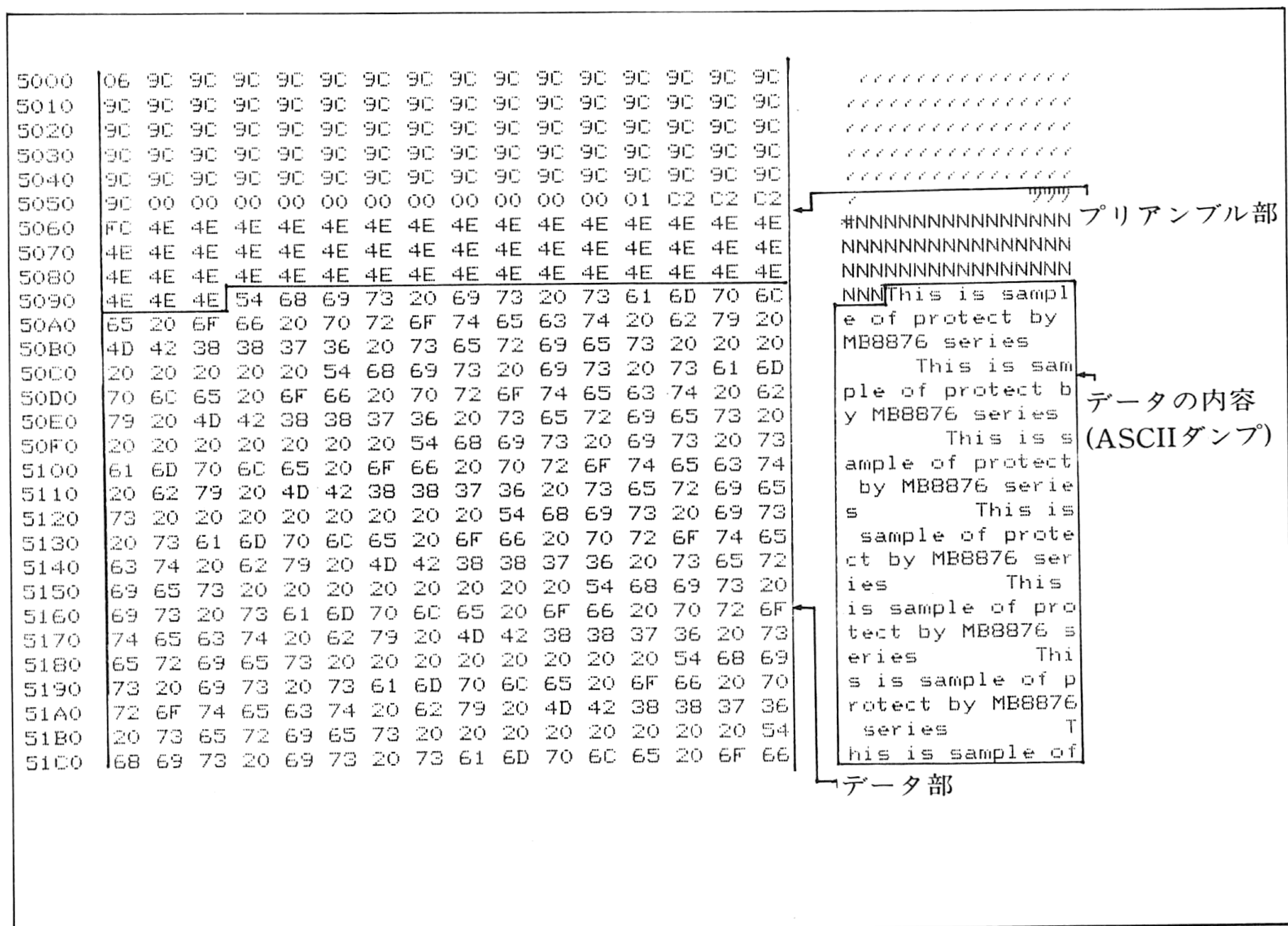
図3-21 にこのフォーマットの2つの例を示す。

いづれもID部が存在しないので，一見アンフォーマットと見ることができる。

また **図3-22** に図 3-21の2つの例を使用したフォーマットのダンプ例を示す。



〔図3-21〕




```
5000  06 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C
5010  9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C
5020  9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C
5030  9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C
5040  9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C
5050  9C 00 00 00 00 00 00 00 00 00 00 00 01 C2 C2 C2
5060  FC 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   #NNNNNNNNNNNNNNN
5070  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   NNNNNNNNNNNNNNNN
5080  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   NNNNNNNNNNNNNNNN
5090  4E 4E 4E 54 68 69 73 20 69 73 20 73 61 6D 70 6C   NNNThis is sampl
50A0  65 20 6F 66 20 70 72 6F 74 65 63 74 20 62 79 20   e of protect by
50B0  4D 42 38 38 37 36 20 73 65 72 69 65 73 20 20 20   MB8876 series
50C0  20 20 20 20 20 54 68 69 73 20 69 73 20 73 61 6D        This is sam
50D0  70 6C 65 20 6F 66 20 70 72 6F 74 65 63 74 20 62   ple of protect b
50E0  79 20 4D 42 38 38 37 36 20 73 65 72 69 65 73 20   y MB8876 series
50F0  20 20 20 20 20 20 20 54 68 69 73 20 69 73 20 73          This is s
5100  61 6D 70 6C 65 20 6F 66 20 70 72 6F 74 65 63 74   ample of protect
5110  20 62 79 20 4D 42 38 38 37 36 20 73 65 72 69 65    by MB8876 serie
5120  73 20 20 20 20 20 20 20 20 54 68 69 73 20 69 73   s        This is
5130  20 73 61 6D 70 6C 65 20 6F 66 20 70 72 6F 74 65    sample of prote
5140  63 74 20 62 79 20 4D 42 38 38 37 36 20 73 65 72   ct by MB8876 ser
5150  69 65 73 20 20 20 20 20 20 20 20 54 68 69 73 20   ies        This
5160  69 73 20 73 61 6D 70 6C 65 20 6F 66 20 70 72 6F   is sample of pro
5170  74 65 63 74 20 62 79 20 4D 42 38 38 37 36 20 73   tect by MB8876 s
5180  65 72 69 65 73 20 20 20 20 20 20 20 20 54 68 69   eries        Thi
5190  73 20 69 73 20 73 61 6D 70 6C 65 20 6F 66 20 70   s is sample of p
51A0  72 6F 74 65 63 74 20 62 79 20 4D 42 38 38 37 36   rotect by MB8876
51B0  20 73 65 72 69 65 73 20 20 20 20 20 20 20 20 54    series        T
51C0  68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 6F 66   his is sample of
```

〔図3-22〕 ①の例

```
5000 |00 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21|  !!!!!!!!!!!!!!!
5010 |21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21|  !!!!!!!!!!!!!!!!
5020 |21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21|  !!!!!!!!!!!!!!!!
5030 |21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21|  !!!!!!!!!!!!!!!!
5040 |21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21|  !!!!!!!!!!!!!!!!
5050 |21 FF FF FF FF FF FF FF FF FF FF FF FE 14 C2 C2|  !            ▒ ﾂﾂ
5060 |FC 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|  #NNNNNNNNNNNNNNN
5070 |4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|  NNNNNNNNNNNNNNNN
5080 |4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E|  NNNNNNNNNNNNNNNN
5090 |4E 4E 4E 00 00 00 00 00 00 00 00 00 00 00 00 14|  NNN
50A0 |A1 A1 FB 54 68 69 73 20 69 73 20 73 61 6D 70 6C|  。。町This is sampl
50B0 |65 20 6F 66 20 70 72 6F 74 65 63 74 20 62 79 20|  e of protect by 
50C0 |4D 42 38 38 37 36 20 73 65 72 69 65 73 20 20 20|  MB8876 series   
50D0 |20 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20|   This is sample 
50E0 |6F 66 20 70 72 6F 74 65 63 74 20 62 79 20 4D 42|  of protect by MB
50F0 |38 38 37 36 20 73 65 72 69 65 73 20 20 20 20 54|  8876 series    T
5100 |68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 6F 66|  his is sample of
5110 |20 70 72 6F 74 65 63 74 20 62 79 20 4D 42 38 38|   protect by MB88
5120 |37 36 20 73 65 72 69 65 73 20 20 20 20 54 68 69|  76 series    Thi
5130 |73 20 69 73 20 73 61 6D 70 6C 65 20 6F 66 20 70|  s is sample of p
5140 |72 6F 74 65 63 74 20 62 79 20 4D 42 38 38 37 36|  rotect by MB8876
5150 |20 73 65 72 69 65 73 20 20 20 20 54 68 69 73 20|   series    This 
5160 |69 73 20 73 61 6D 70 6C 65 20 6F 66 20 70 72 6F|  is sample of pro
5170 |74 65 63 74 20 62 79 20 4D 42 38 38 37 36 20 73|  tect by MB8876 s
5180 |65 72 69 65 73 20 20 20 20 54 68 69 73 20 69 73|  eries    This is
5190 |20 73 61 6D 70 6C 65 20 6F 66 20 70 72 6F 74 65|   sample of prote
51A0 |63 74 20 62 79 20 4D 42 38 38 37 36 20 73 65 72|  ct by MB8876 ser
51B0 |69 65 73 20 20 20 20 54 68 69 73 20 69 73 20 73|  ies    This is s
51C0 |61 6D 70 6C 65 20 6F 66 20 70 72 6F 74 65 63 74|  ample of protect
```

プリアンブル部

データの内容(ASCIIダンプ)

データ部

□はSYNC及びDAM部を示す

〔図3-22〕 ②の例

3-13　GAPにデータを入れるプロテクト

■8876

通常，GAPはスプライシングギャップを避けるなどに使用されるエリアで，このエリアにデータを書き込むことはない。

このプロテクトは，逆に通常データ領域として使用されていないこのエリアにデータを書き込むことによって作成するプロテクトである。

GAPに関しては，任意のデータを書き込んでもリード，ライト動作には問題がないが，SYNCには本来の値（8876では 00_H または FF_H，765では 00_H のみ）以外が書き込まれると，アドレスマークをサーチすることが不可能となるので，SYNCにはデータを書き込むことは出来ない。

注意点としては，GAPに任意のデータが書き込めるのは，フォーマットを行なう場合のみである。

また，このプロテクトをかけたトラックのセクターにはライト動作を行なってはならない。

これは，ライト動作によりGAP3のデータが破壊される（スプライシング・ギャップが発生する）ためである。

リード動作，及びリードアドレスなどは，行なって良い。

チェックの方法は，リード・トラックにより，1トラック分のデータを読み出し，GAPのデータをチェックすれば良い。

図3－23に例を示す。

なお，このプロテクトは，765でも読み出すことが可能（リード・ダイアグノスティックを使用する）なので，8876でこのフォーマットを行ない，765用のプロテクトとして使用することも可能である。

また，このプロテクトは765では外すことは出来ない。

```
5000 | F8 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C |  〒ﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜ
5010 | 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C |  ﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜ
5020 | 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C |  ﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜ
5030 | 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C |  ﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜ
5040 | 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C |  ﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜﾜ
5050 | 9C 00 00 00 00 00 00 00 00 00 00 00 01 C2 C2 C2 |  ﾜ            ﾂﾂﾂ
5060 | FC 73 61 6D 70 6C 65 20 6F 66 20 73 61 6D 70 6C |  #sample of sampl
5070 | 65 20 6F 66 20 73 61 6D 70 6C 65 20 6F 66 20 73 |  e of sample of s      ← GAP1
5080 | 61 6D 70 6C 65 20 6F 66 20 73 61 6D 70 6C 65 20 |  ample of sample
5090 | 6F 66 20 00 00 00 00 00 00 00 00 00 00 00 00 14 |  of
50A0 | A1 A1 FE 00 00 01 01 FA 0C 70 72 6F 74 65 63 74 |  ｡｡■    ▣ protect
50B0 | 65 64 20 62 79 20 4D 42 38 38 37 36 20 20 20 00 |  ed by MB8876          ← GAP2
50C0 | 00 00 00 00 00 00 00 00 00 00 00 14 A1 A1 FB FF |             ｡｡町
50D0 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
50E0 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
50F0 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5100 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5110 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5120 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5130 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5140 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |  ←データ部
5150 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5160 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5170 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5180 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5190 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
51A0 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
51B0 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
51C0 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FB |  GAP3             町
51D0 | E5 4D 42 38 38 37 36 20 20 20 20 4D 42 38 38 37 |  ■MB8876    MB887
51E0 | 36 20 20 20 20 4D 42 38 38 37 36 20 20 20 20 4D |  6    MB8876    M
51F0 | 42 38 38 37 36 20 20 20 20 00 00 00 00 00 00 00 |  B8876
5200 | 00 00 00 00 00 14 A1 A1 FE 00 00 02 01 AF 5F 70 |       ｡｡■    ｯ_p
5210 | 72 6F 74 65 63 74 65 64 20 62 79 20 4D 42 38 38 |  rotected by MB88
5220 | 37 36 20 20 20 00 00 00 00 00 00 00 00 00 00 00 |  76
5230 | 00 14 A1 A1 FB FF FF FF FF FF FF FF FF FF FF FF |    ｡｡町
5240 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5250 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
5260 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |  ←次のセクター
5270 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
```

□ はGAPに書き込んだデータを示す。(ASCIIダンプを参照)

〔図3-23〕

第 4 章

同一FDCでは外せないプロテクト

はじめに

第３章中でも，いくつか御紹介したが，EDC の違いから，「同一 FDC では，そのプロテクトがかかっていることのチェックは行なえるが，同じフォーマットを作成することが出来ない，つまり同一 FDC はプロテクトを外すことが不可能なプロテクト」をこの章では中心として扱う。

あるホストマシンのユーザーが，そのマシンにより外すことが出来ないプロテクトをかければ，市場でのコピーなどの確率は，かなり減少するものと考えられる。

また，FDC の異なる２台のマシンを持っているユーザーでも，同一ディスケット内のあるトラックには，１台目のマシンでは外すことの出来ないプロテクトをかけ，別のトラックには，２台目のマシンでは外すことの出来ないプロテクトをかければ，さらにコピーされる確率が低くなる。

おそらく，将来はこの手法のプロテクトが多くなると考えられるが，逆に，商品（もちろん，ソフトウェア）の量産時にも，２種類以上のマシンが必要となり，さらにオペレーターの人員が多く必要となるので，工数的，経費的に問題が生ずる場合もある。

この章では，今まで解説に使用した，NEC のμPD765シリーズと富士通の MB8876 シリーズ（この章でも，それぞれ，765，8876と略させていただく）相互間で互いに自分自身では外すことのできないプロテクトを紹介する。

なお，765は，8876のライト・トラックコマンドに相当するコマンドがないため，765自身では外せないプロテクトのほうが多くなることを知っていただきたい。

4-1　8876では外せないプロテクト

8876で「フォーマット出来ない点，及び765と異なった動作をする点は，

1.　フォーマッティング時にF5H～FEHまでのデータが，特別なコントロールデータとして使用されるため，その値そのものが書き込めない場合がある。
2.　ID部のデータ長Nの扱いが765と異なり，下位2ビットのみしか使用されないので，リード，ライト時には，N＝3（1024バイト／セクター）までしか使用出来ない。

の2つである。

1について補足すると，たとえば，ライト・トラックコマンドを実行する（MFMモード）場合，8876に以下のデータが送られると，次のような動作が行なわれる。

F5H：ミッシング・クロックにてA1Hを書き込む。
F6H：ミッシング・クロックにてC2Hを書き込む。
F7H：内部で計算されたCRC2バイトを書き込む。

よって，F5H，F6H，F7Hの各データそのものは，フォーマット時に使用できないことになる。

2については，第3章にて解説したとうりである。

さて，上記2つの特性を使用して，MB8876では外せないプロテクトを次に紹介する。

なお，プロテクトをかけるFDCには765を使用する。

〔補足〕

8876を使用したシステムの場合，ハード的にMFM，またはFMモードに固定されることがある。

この場合，たとえばMFMモードに固定されている場合，FMフォーマットにて，フォーマット及びライト動作が行なえないが，同時にリード動作も行なえないために，この点をプロテクトとして有効に使用することは不可能である。

（しかし，ハード的に固定されているモードではないモードで，765などでフォーマットを行なえば，擬以アンフォーマットとして使用することは可能である。）

4-2 フォーマット時に使用できないデータを使用したプロテクト

8876がフォーマット時に，F5H，F6H，F7H（MFM時の場合。FM時は異なる）が書き込めないことを利用するプロテクトである。

たとえば，

1. フォーマットした後，変更できないパラメータ部に，上記のデータが存在する。
2. フォーマットのみがされている（データが書き込まれていない）状態（このチェックには，第3章のスプライシング・ギャップを利用したプロテクトのチェック法を利用する）にて，データ部が，上記のデータでフォーマットされている。

の2つの例では，765でフォーマットすることが可能で，かつ，8876自身では外すことの不可能なプロテクトである。

それぞれについて，具体例を示す。

1.の例

フォーマット後，変更できないパラメータ部として最適なのが，ID部のパラメータである。

ID部のパラメータの内，Nを除くC，H，Rについては，自由に可変できる（第2章参照）ので，この各パラメータに，8876で書き込めないデータを使用すれば良い。

図4-1に，765でこのフォーマッティングを行なった例のIDダンプを示す。

もちろん，各セクターは，8876でもリード，ライト可能であるので，扱いとしては何ら変らない。

2.の例

データ部に8876ではフォーマット不可能なデータを使用してフォーマットを行なえば良い。

例として，**図4-2**に，766でこのフォーマッティングを行なった例を示す。

なお，このダンプリストは，8876のリード・トラックにより読み出したものである。

チェックの方法としては，図のデータ部が，すべてF7H（8876でフォーマット出来ない）であることのチェックと，GAP3のデータがすべて4EH（フォーマット時のデータ，この値なら，スプライシング・ギャップが発生していないと判断する）であることを確認する。

場合によっては，より厳密性を増すため，SXHC部などのデータのチェックも必要である。

```
ID dump program
drive No.? 1
track No.? 0
How many sectors? 16
NO.          C             H             R             N
 1           F5            F5            F5            1
 2           F6            F6            F6            1
 3           F7            F7            F7            1
 4           F5            F6            F7            1
 5           0             0             5             1
 6           0             0             6             1
 7           0             0             7             1
 8           0             0             8             1
 9           0             0             9             1
 10          0             0             A             1
 11          0             0             B             1
 12          0             0             C             1
 13          0             0             D             1
 14          0             0             E             1
 15          0             0             F             1
 16          0             0             10            1
Hard copy? y
```

このセクターのIDのパラメータに8876ではフォーマット不可能な，F5H，F6H，F7Hが使用されている。

〔図4-1〕

5000	09	09	09	09	09	09	09	09	00	84	84	84	84	84	84	84	
5010	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	
5020	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	←プリアンブル部
5030	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	
5040	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	
5050	84	84	84	84	84	84	84	87	FF	FF	FF	FF	FF	FF	FF	FF	
5060	FF	FF	FF	F8	14	C2	C2	FC	4E	4E	4E	4E	4E	4E	4E	4E	
5070	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5080	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5090	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	
50A0	00	00	00	00	00	00	14	A1	A1	FE	00	00	01	01	FA	0C	←ID部
50B0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
50C0	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	
50D0	00	00	14	A1	A1	FB	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
50E0	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
50F0	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	データ部，先頭のSYNC, DAMを除く実際のデータの部分が，すべてF7H(8876でフォーマット不可能な値)であることをチェックする。
5100	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
5110	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
5120	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
5130	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
5140	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
5150	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
5160	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
5170	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
5180	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
5190	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
51A0	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
51B0	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
51C0	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	
51D0	F7	F7	F7	F7	F7	F7	1A	83	4E	4E	4E	4E	4E	4E	4E	4E	
51E0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	GAP3，すべて4EHであることを確認する。
51F0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5200	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	
5210	00	00	00	00	00	00	00	14	A1	A1	FE	00	00	02	01	AF	
5220	5F	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5230	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	←次のセクター
5240	00	00	00	14	A1	A1	FB	F7	F7	F7	F7	F7	F7	F7	F7	F7	
5250	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	F7	

〔図4-2〕

4-3 ID部のN(データ長)の扱いの違いによるプロテクト

8876は，前述のように1024バイト／セクター以上のデータ長のセクターに対してリード，ライトが行なえない。

しかし，リード・トラックにより，それ以上のデータ長のセクターの内容を読み出すことが可能であるが，そのデータ中に，フォーマット時に書き込めない $F7_H$ などのデータがある場合は，ライト・トラックにて同じ内容をフォーマット時に書き込むことは不可能となる。

よって，例として次のようなプロテクトが考えられる。

まず，8876または，765により，N＝4（2048バイト／セクター）またはN＝5（4096バイト／セクター）のデータ長フォーマットを行なう。

また，フォーマット時のIDのNの値も実際にフォーマットされるデータ長の値と同じにしておく。

次に，実際にフォーマットしたデータ長分のデータ，またはプログラム（これらは，ダミーのデータでも，実際使用されるものでもどちらでも良い）を用意する。

ただし，このデータ，または，プログラム内には，必ず8876でフォーマットする場合に書き込めない $F7_H$ などのデータを含んでいることが必要である。

また，データまたはプログラムがダミーで良ければ，すべて $F7_H$ などの単一データでも良い。この場合は，フォーマット時に，765でイニシャライズデータとして $F7_H$ などを指定すれば，上記のデータを用意する必要はない。

つまり，フォーマット時にデータ部がイニシャライズデータで埋められるためである。

さて，データ又はプログラムを用意したら，これを765によって，セクターに書き込む。

765の場合は，Nの値と処理されるデータのバイト数が一致しているので，2048バイトまたは4096バイトが正常に書き込まれる。

これでプロテクトが完了した。

次にプロテクトのチェック法について示す。

まず，リードデータ・コマンドにより，プロテクトのかかったセクターのデータを読み出す。

この時，読み込まれるデータのバイト数は，N＝4の時128バイト（N＝0と等価）N＝5の時256バイトである。

また，データ読み込み終了時にデータ部にCRCエラーが発生していることを確認する(発生しない場合，データの内容を変更する)。

次にリード・トラックコマンドにより，このセクターの全内容（N＝4の時2048バイト，N＝5の時4096バイト）を読み出し，このデータが，正しいものであるかどうかをチェックする。

また，この読み出したデータがプログラムであれば，実行を行なう。

この場合，読み出されたプログラムが正しくない場合，プログラムが正常に動作せず（時には暴走することも考えられる）次にステップへ進めないことになる。

さて，ここまでお分りと思うが，リードデータコマンドで読み取られたデータ中，もしくは，読み取れない残りのデータ中のいずれかに $F7_H$ など8876ではフォーマットが不可能であるデータがあれば良いわけである。

つまり，8876ではフォーマットが不可能なデータがリードデータで読み取れた部分にのみある場合は，フォーマット時にこのデータを「避けて」フォーマットを行ない，ライトデータによって書き込むことが考えられるが，この場合，データ部のCRCがエラーとならないことと，書き込みの際，スプライシング・ギャップが発生するので，リード・トラックにてデータを読み出した場合，データが，ビットずれを起こし，データが正確に読めない。また，リードデータによって読み取れないデータの部分（当然ライトデータコマンドによっても書き込めない）に，フォーマット不可能のデータがある場合は，同じフォーマッティングを行なうこと自体，8876では不可能である。

以上で8876では外すことのできない強力なプロテクトが完成する。

図4-3にこのプロテクトをN＝4で作成した場合の例を示す。

8876でフォーマットが不可能なデータがどの位置にあるかに注目していただきたい。

5000	21	21	21	21	21	21	21	21	13	93	93	93	93	93	93	93
5010	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93
5020	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93
5030	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93
5040	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93
5050	93	93	93	93	93	93	93	80	00	00	00	00	00	00	00	00
5060	00	00	00	30	C2	C2	FC	4E	4E	4E	4E	4E	4E	4E	4E	4E
5070	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
5080	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
5090	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00
50A0	00	00	00	00	00	14	A1	A1	FE	00	00	01	04	AA	A9	4E
50B0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
50C0	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	00
50D0	00	14	A1	A1	FB	FF	54	68	69	73	20	69	73	20	73	61
50E0	6D	70	6C	65	20	6F	66	20	70	72	6F	74	65	63	74	2E
50F0	20	43	61	6E	27	74	20	63	6F	70	79	20	62	79	20	4D
5100	42	38	38	37	36	20	73	65	72	69	65	73	FF	FF	FF	FF
5110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5120	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5140	FF	FF	FF	FF	FF	F5	F6	F7	FF	FF	FF	FF	FF	FF	FF	FF
5150	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5160	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5170	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5180	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5190	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
51A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
51B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
51C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
51D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
51E0	FF	FF	FF	FF	FF	F5	F6	F7	FF	FF	FF	FF	FF	FF	FF	FF
51F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5200	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5210	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5220	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5230	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5240	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5250	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5260	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5270	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5280	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5290	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
52A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
52B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
52C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
52D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
52E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
52F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5300	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

←プリアンブル部

← ID部，IDは C＝0 0 H　H＝0 0 H　R＝0 1 H　N＝0 4 H

SYNC及びDAM

このF5H，F6H，F7Hは，8876ではフォーマット不可能なデータ

←この点線までの 128 バイトがリードデータによって読み取られるデータ部

この 2 バイトがリードデータで読み取ったときのデータ部の CRC に相当する。この値がエラーとなるように調整する。

リードデータで読み取れない部分に存在する8876でフォーマット不可能のデータ。

5310	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5320	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5330	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5340	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5350	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5360	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5370	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5380	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5390	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
53A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
53B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
[illegible]			[illegible]	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	[illegible]		
						[illegible]	FF	FF	FF	FF	FF	[illegible]				
5850	FF	[illegible]	[illegible]	[illegible]	[illegible]	FF	FF	FF	FF	FF	FF	FF	FF	[illegible]		
5860	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5870	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5880	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5890	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
58A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
58B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
58C0	FF	FF	FF	FF	FF	F5	F6	F7	FF	FF	FF	FF	FF	FF	FF	FF
58D0	FF	FF	FF	FF	FF	71	02	4E	09	C9	C9	C9	C9	C9	C9	C9
58E0	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9
58F0	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9	C9
5900	C9	C9	C9	C9	C9	C9	C9	C9	C0	00	00	00	00	00	00	00
5910	00	00	00	00	14	A1	A1	FE	00	00	02	04	FF	FA	4E	4E
5920	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
5930	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	00	00
5940	14	A1	A1	FB	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	F

← データ部

「8876ではフォーマット不可能なデータ。このようにデータ部内にいくつか存在させるとより効果的である。

← GAP3

← 次のデータ

765でN＝4でデータを書き込んだ時のデータ部のCRC。データ部のみでなく，この値もチェックすると良い。

〔図4-3〕

4-4　765では外せないプロテクト

765で「フォーマット出来ない」点，及び8876と異なった動作をする点は，

1. フォーマッティング時に可能なフォーマットは，IBM3740フォーマット（FM時）及び，IBMシステム34フォーマット（MFM時）に準ずるものだけである。

2. ID部のデータ長Nの扱いが8876と異なり，N＝6（8192バイト／セクター）まで使用可能（マニュアルでは，この値までしか規定されていない）で，Nの値の下位2ビットのみ有効とする8876とは，N＝4以上でリード，ライト動作が異なる。

2は，8876の「ID部のNの扱いの違いによるプロテクト」と同様なプロテクトをかけることが可能である。

1はIBMフォーマット，またはECMAフォーマット以外のフォーマットを，8876などで作り，使用するプロテクトが考えられるが，これは強力で，765では，いかなる方法でも外せないプロテクトをかけることが可能となる。

また，このプロテクトは，バリエーションが多く考えられる。

4-5　ID部のN(データ長)の扱いの違いによるプロテクト

8876でも同様のプロテクトの紹介をしたが，今度は逆に同様な手法で765用のプロテクトを考える。

注意点は，プロテクトのチェック方法が8876の場合と異なる点でる。

以下に具体例を示しながら解説する。

まず，対象となるセクターのデータ長は，8876の場合と同様にN＝4，5（5インチ，MFMの場合）となる。

初めに，N＝4，ないしはN＝5にて，765にてフォーマットを行なう。もちろん実際のデータ長も同じ長さとする。

次に，8876にて，N＝4，またはN＝5でデータの書き込みを行なう。

この場合，8876では，N＝4はN＝0，N＝5はN＝1と等価として扱われるため，N＝4の場合は，データ部の128バイト分のみ，N＝5の場合はデータ部の256バイト分のみが書き込まれる。

この書き込み後，765にてリード動作を行なうと，データに読み取れるが，異なるデータ長にて8876でデータの書き込みを行なったため，データ部のCRCエラーが発生する。

しかし，8876で書き込みを行なった128バイト分，ないしは256バイト分のデータは保障されているので，この内容をチェックまたは使用することが可能である。

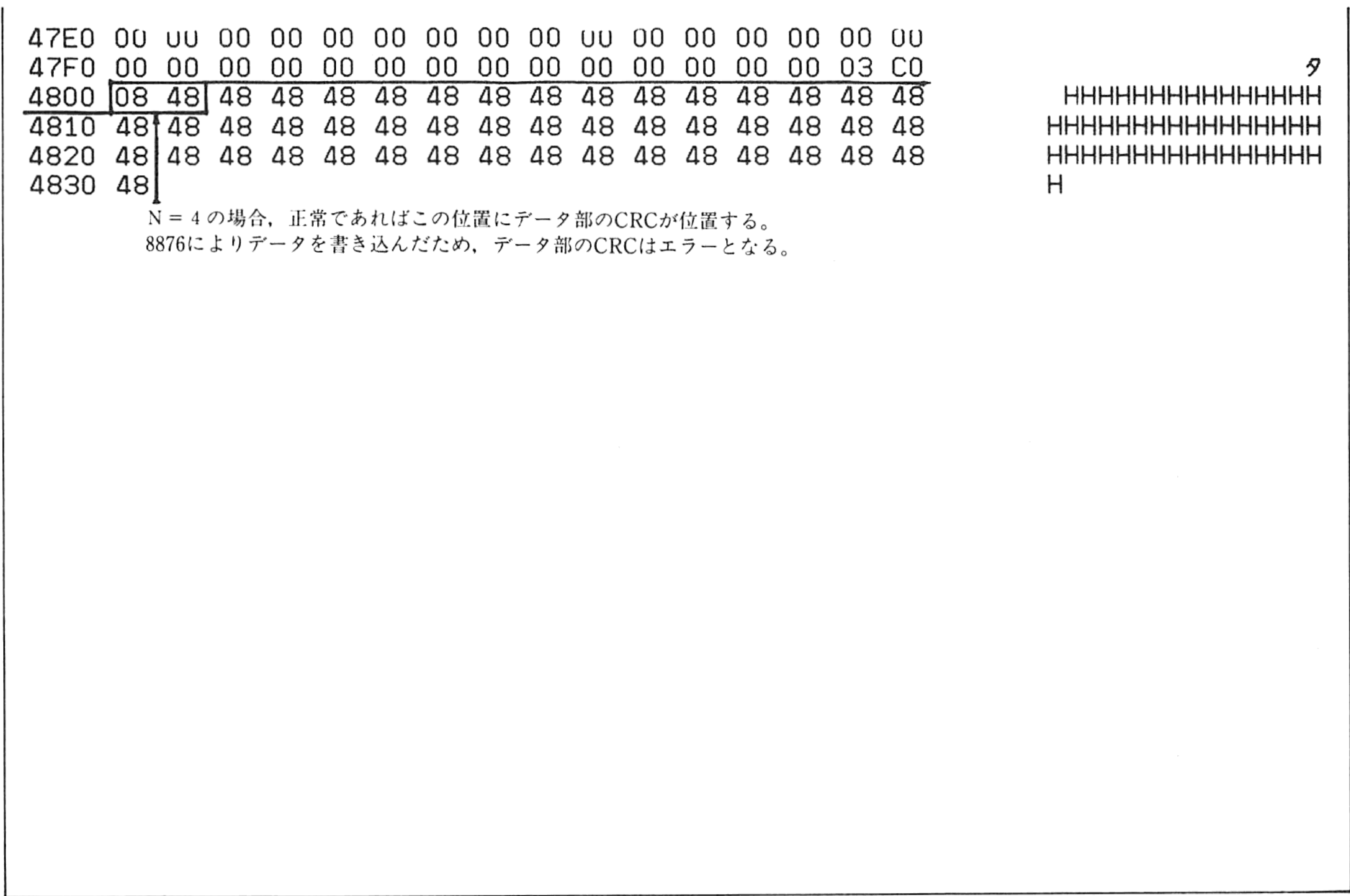
さて，同様フォーマットを765のみで行なおうとした場合，一旦データを書き込むと，データ部のCRCが正常に書かれてしまうので，不可能であり，フォーマット時に異なるデータ長（つまりID部のNの値と実際フォーマットするデータ部のデータ長を変える）を使用すれ

4000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4080	77	BE	FE	00	00	00	00	00	00	00	00	00	00	00	00	00
4090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
41F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4240	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4250	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4270	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4300	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4310	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4330	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4360	00	00	00	00	00	00	00	00	00	[illegible]	[illegible]	[illegible]	00	00	00	00
4370	00	00	00	00	00	00	00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	00	00
[illegible]	[illegible]	[illegible]	00	00	00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

← 8876で書き込んだ128バイトのデータ

← 8876で書き込んだ場合のデータ部のCRC

← フォーマット時のデータそのまま。本来FFHであるが，8876で書き込んだため，ビットずれを発生して00Hとなっている。



```
47E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
47F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 C0                 タ
4800 08 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48   HHHHHHHHHHHHHHHH
4810 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48   HHHHHHHHHHHHHHHH
4820 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48   HHHHHHHHHHHHHHHH
4830 48                                                H
```


〔図4-4〕

ば，確かに，データ部のCRCエラーが発生するが，逆にデータが書き込まれていないので，読み出した場合のチェックで，異なることが発見される。

よって，765だけでは，このフォーマットは不可能であることが理解できる。

図 4-4 にこの方法にてかけたプロテクト例の，データ部のダンプリストを示す（N＝4の場合)。

このデータからはわからないが，読み取り時にデータ部のCRCエラーが発生している。

4-6 データ部を存在させずIDのみのセクタを増やすことによるプロテクト

1トラックの容量は規格では6250バイト（5インチ，MFM時）となっている。

よって，この値から1トラック内に入るセクター数が決定される。

しかし，データ部がなく，ID部のみのセクターを作成することが可能であれば，その数はさらに多くなる。

765でフォーマットを行なう場合，プリアンブル部が146バイト，N＝0，GAP3＝1の場合，1セクターあたり191バイトとなるので，1トラックに31セクター入る。

ところが，ID部のみのセクターを作成することが可能なら，1セクターあたり44バイトであるので，1トラックに138セクター入ることになる。

つまり，8876によって32セクター以上のIDのみのフォーマットを行ない，765のリードIDコマンドによってすべてのIDが存在しているかどうかのチェックを行なう。

765の場合，1トラック内に32以上セクターが存在するようなフォーマットは不可能であるので，このプロテクトは765では外せないことになる。

具体的にどのようなフォーマットになるかは第3章の図3-15を参照されたい。

図4-5に，このフォーマットによってプロテクトをかけたトラックのIDダンプ）この例では1トラック内に50個のID部のみセクターが存在する）及び，**図4-6**に8876のリード・トラックにより読み取ったトラックの最初の部分を示す。

```
ID dump program
drive No. 1
track No. 0
How many sectors 50
NO.            C            H            R            N
 1             0            0            1            1
 2             0            0            2            1
 3             0            0            3            1
 4             0            0            4            1
 5             0            0            5            1
 6             0            0            6            1
 7             0            0            7            1
 8             0            0            8            1
 9             0            0            9            1
 10            0            0            A            1
 11            0            0            B            1
 12            0            0            C            1
 13            0            0            D            1
 14            0            0            E            1
 15            0            0            F            1
 16            0            0            10           1
 17            0            0            11           1
 18            0            0            12           1
 19            0            0            13           1
 20            0            0            14           1
 21            0            0            15           1
 22            0            0            16           1
 23            0            0            17           1
 24            0            0            18           1
 25            0            0            19           1
 26            0            0            1A           1
 27            0            0            1B           1
 28            0            0            1C           1
 29            0            0            1D           1
 30            0            0            1E           1
 31            0            0            1F           1
 32            0            0            20           1
 33            0            0            21           1
 34            0            0            22           1
 35            0            0            23           1
 36            0            0            24           1
 37            0            0            25           1
 38            0            0            26           1
 39            0            0            27           1
 40            0            0            28           1
 41            0            0            29           1
 42            0            0            2A           1
 43            0            0            2B           1
 44            0            0            2C           1
 45            0            0            2D           1
 46            0            0            2E           1
 47            0            0            2F           1
 48            0            0            30           1
 49            0            0            31           1
 50            0            0            32           1
Hard copyy
```

〔図4-5〕

5000	F8	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	
5010	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	
5020	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	→ プリアンブル部
5030	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	
5040	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	9C	
5050	9C	00	00	00	00	00	00	00	00	00	00	00	01	C2	C2	C2	
5060	FC	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5070	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5080	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5090	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	00	00	14	
50A0	A1	A1	FE	00	00	01	01	FA	0C	4E	4E	4E	4E	4E	4E	4E	→ 1番目のセクター
50B0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	
50C0	00	00	00	00	00	00	00	00	00	00	00	14	A1	A1	FE	00	→ 2番目のセクター
50D0	00	02	01	AF	5F	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
50E0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	
50F0	00	00	00	00	00	00	00	14	A1	A1	FE	00	00	03	01	9C	→ 3番目のセクター
5100	6E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5110	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	
5120	00	00	00	14	A1	A1	FE	00	00	04	01	05	F9	4E	4E	4E	→ 4番目のセクター
5130	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5140	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	00	00	14	
5150	A1	A1	FE	00	00	05	01	36	C8	4E	4E	4E	4E	4E	4E	4E	→ 5番目のセクター
5160	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	
5170	00	00	00	00	00	00	00	00	00	00	00	14	A1	A1	FE	00	→ 6番目のセクター
5180	00	06	01	63	9B	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5190	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	
51A0	00	00	00	00	00	00	00	14	A1	A1	FE	00	00	07	01	50	→ 7番目のセクター
51B0	AA	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
51C0	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	
51D0	00	00	00	14	A1	A1	FE	00	00	08	01	40	94	4E	4E	4E	→ 8番目のセクター
51E0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
51F0	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	00	00	14	
5200	A1	A1	FE	00	00	09	01	73	A5	4E	4E	4E	4E	4E	4E	4E	→ 9番目のセクター
5210	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	
5220	00	00	00	00	00	00	00	00	00	00	00	14	A1	A1	FE	00	→ 10番目のセクター
5230	00	0A	01	26	F6	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5240	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	
5250	00	00	00	00	00	00	00	14	A1	A1	FE	00	00	0B	01	15	→ 11番目のセクター
5260	C7	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
5270	4E	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	
5280	00	00	00	14	A1	A1	FE	00	00	0C	01	8C	50	4E	4E	4E	→ 12番目のセクター
5290	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	
52A0	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	00	00	14	
52B0	A1	A1	FE	00	00	0D	01	BF	61	14	12	12	12	12	12	12	→ 13番目のセクター
52C0	12	12	12														

各セクターはID部のみしか存在しない。

〔図4-6〕

4-7　1トラック内で完全な形でデータ長の異なるセクターを存在させるプロテクト

1トラック内に異なるデータ長のセクターを混在させるプロテクトについては，第3章で解説した。

しかし，765では，完全な形で(つまり，フォーマットにてセクターごとのデータ長を変化させる）は，データ長の異なるセクターを存在させるようなフォーマットは不可能である。

そこで，8876にて1セクター内に異なるデータ長のセクターを完全な形で混在させるようなフォーマットを行ない，765のプロテクトとして使用するのがこの方法である。

具体例を**図4-7**に示す。

この図では，1セクター目が256バイト／セクター，2セクター目が512バイト／セクターとなっている。

また，この例では単にこのフォーマットを行なっているだけでデータは書き込まれていないが，もちろん，リード，ライトは正常に可能であるので，通常のセクターと同様に扱うことが可能である。

チェック方法としては，リード・ダイアグノスティックにて，1トラック分のデータを読み出し（5インチ，MFM時はN＝6で読み出すと8192バイト読み出すことが可能であるので，トラック1周分6250バイト以上読み出すことが可能である）各セクターのデータ長が正しいかどうかをチェックすれば良い。

特に注意してチェックしなければならないのは，各セクターのデータ部後がすぐにGAPになっているかどうかのチェックである。

第3章で紹介したように,765でデータ長の異なるセクターを混在させるフォーマットを作成すると，必ずデータ部後,「ゴミ」が存在するので，この「ゴミ」が存在すれば，765にてフォーマットを行なったと確認できるためである。

4000	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4100	FB	E5	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
4110	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
4120	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
4130	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	00
4140	00	A1	A1	A1	FE	00	00	02	02	9F	3C	4E	4E	4E	4E	4E
4150	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E
4160	4E	00	00	00	00	00	00	00	00	00	00	00	00	A1	A1	A1
4170	FB	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4180	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4190	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
41A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
41B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
41C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
41D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
41E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
41F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4200	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4210	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4220	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4230	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4240	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4250	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4260	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4270	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4280	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4290	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
42A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
42B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
42C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
42D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
42E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
42F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4300	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4310	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4320	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4330	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4340	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

CRC (→ 4100: FB E5)

1番目のセクターのデータ部
データ長は256バイト/セクター

GAP3

2番目のセクターのID部

2番目のセクターのデータ部
データ長は512バイト/セクター

```
 ◣NNNNNNNNNNNNNN
NNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNN
NNNNN
 。。。        ノくNNNNN
NNNNNNNNNNNNNNNN
N              。。。
```

4350	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
4360	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
4370	FF	A5	CF	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	・マNNNNNNNNNNNNN
4380	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	NNNNNNNNNNNNNNNN
4390	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	NNNNNNNNNNNNNNNN
43A0	4E	4E	4E	4E	4E	4E	00	00	00	00	00	00	00	00	00	00	NNNNNN
43B0	00	00	A1	A1	A1	FE	00	00	03	01	9C	6E	4E	4E	4E	4E	｡｡｡ ｍNNNN
43C0	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	4E	NNNNNNNNNNNNNNNN
43D0	4E	4E	00	00	00	00	00	00	00	00	00	00	00	00	A1	A1	NN ｡｡
43E0	A1	FB	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	｡
43F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
4400	FF																

CRC (→ A5 CF)
GAP3
3番目のセクター

〔図4-7〕（765のリードダイアグノスティックを使用）

第 5 章

プロテクトを持つ プログラムの開発

前章までは，プロテクトそのものの内容について解説を行なってきたが，この章では，プロテクトをかけたプログラム（ディスケット，または商品とも言いかえられる）を開発するにあたってのポイント，注意点などについて解説する。

5-1　プロテクトのチェック方法
ランダムテクニック

いくらプロテクトそのものが強力であっても，チェックの方法，手順が貧弱であれば，その効果は半減してしまう。

プロテクトをかける場合，プロテクト自体の強さよりも，チェック方法の強さのほうが重要であるとも考えられる。

むしろ，プロテクト自体がそれほど強力でないにしても，チェック方法が強力であれば，商品全体としては強力なプロテクトがかかっているのと等価と見ることができるほどである。

さて，プログラム中のどこでこのプロテクトのチェックを行なうかであるが，これはプログラム自体の構成にもよるが，プログラムが起動されてからの「流れ」の各節でチェックするのが一般的である。

一般的なプログラムの起動からの流れは，

1　BOOT 後，IPL を読み込む（IPL がホストマシン内に固定されている場合もある。

2　IPL により，第 2 IPL，ないしはメインプログラムまたはメインのプログラムモジュールをロードし，実行する．

3　第 2 IPL がある場合は，これによりメインプログラムなどを読み込み実行する．

と言った形をとる。

プロテクトをチェックする最初の節としては，IPL によりチェックをする方法がある。

IPL でチェックを行なう場合，IPL の物理的なディスケット上の位置は，ホストマシンによって固定されており，該当するセクターを読み取り解析されると，簡単にプロテクトを外されてしまう可能性がある（かつ，IPL 自身にはプロテクトをかけることは不可能である場合が多い)。

よって，これを避けるためには，IPL をいくつかに分けて続くプログラム，データをロードして行く方法が考えられる。

つまり，最初の IPL により，第 2 IPL をロードし，第 2 IPL により，第 3 IPL をロード………と言った方法で，IPL を次々にロードし，かつ各 IPL によってプロテクトのチェックを行う。

また，この手法だと，第 2 IPL 以後にはプロテクトをかけても良いので，さらに効果的である。

IPLにより，プログラムのメインモジュールを読み込むが，このプログラムモジュールをいくつかに分けてロードする方法が考えられる。

もちろん，分けてロードする場合は，それぞれの部分でさらにプロテクトをチェックすると良い。

ここで注意しなければならないのは，分割してロードを行なうと（IPLも含めて）それだけ起動までの時間が長くなる，と言う点である。

よって何らかのプログラミング上のテクニックが必要となることを，考えていただきたい。

メインプログラム，またはモジュールが起動した後は，その内部でプロテクトのチェックを行なえば良い。

メインプログラムから，各モジュールを呼んで動作させるようなタイプ（ビジネス・ソフトに多い）では，各モジュールを読み込むごとに，その内部で異なったプロテクトのチェックを行なうと良い。

これは，メインプログラム起動までは正常に動いても一部のユーティリィティーのみが動かなくすることが可能となり，全体として，コピー品が役に立たなくする手としてよく使用される。

次に，プロテクトをかけるトラック，またはセクターについて考察する。

どのトラックにプロテクトをかけるのが効果的であるかであるが，これは，プログラムの構成によって異なるが，一般的には一ケ所に集中させず，バラバラに置くと良い。

ランダムに置くことにより，プロテクトを解析し，外すと言った手順が難しくなるためである。

ランダムに置く場合，「一見プロテクトがかかっていないように見えるプロテクト」（第3章参照）を使用すると効果的である。

プロテクトのかかっているトラックを捜すこと自体，大変難しくなるためである。

また，良く使用されるトラックとして，IPLの存在するトラックがある。

IPLが入っているセクターには，プロテクトをかけることは不可能と思えるが，ホストマシンのIPL読み込み処理しだいで，プロテクトをかけることも可能である。

これについては，ホストマシンごとに異なるので，それぞれについて解析していただきたい。

また，IPLが入っているセクター以外の同一トラック上のセクターに対してはプロテクトが可能である。

IPLの存在するトラックにプロテクトをかけると次のような利点がある。

1　IPLと同一トラック上にプロテクトがかかっているため，ブート後，プロテクトのチェックのために他のトラックにシークする必要がなく起動が速い。

2　IPLのトラック＝0トラック，ないしは1トラックの場合が多い。

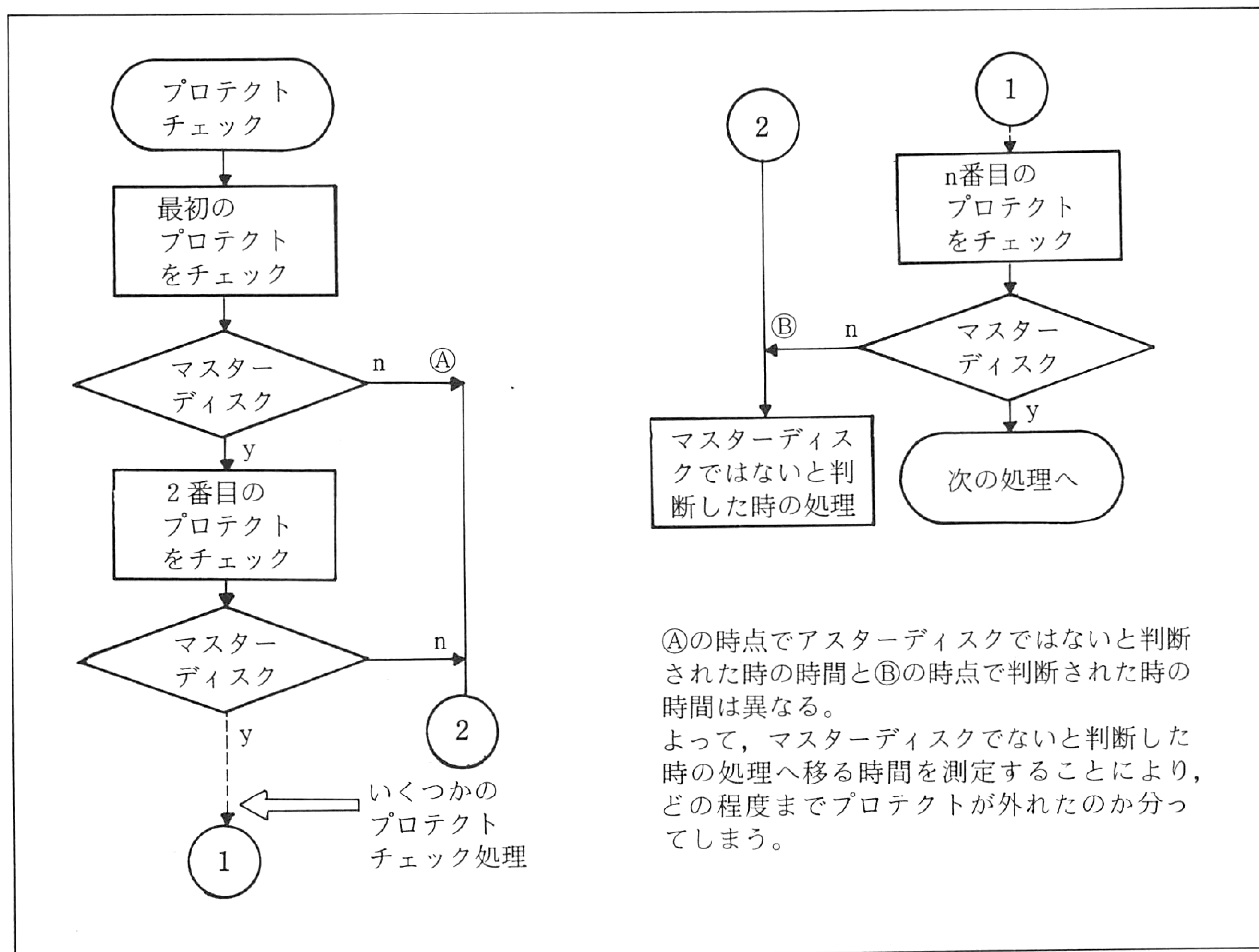
バックアップユーティリティーでは，0トラックからバックアップを行なうものが多く，IPLのトラックにプロテクトがかかっている場合，最初のトラックでエラーが発生し，コピーさせないようにすることが可能である。

3　IPLのトラックのみにプロテクトをかけると，他の一般的に言われているユーザーエリアはフリーで，プログラム上の制約がない。

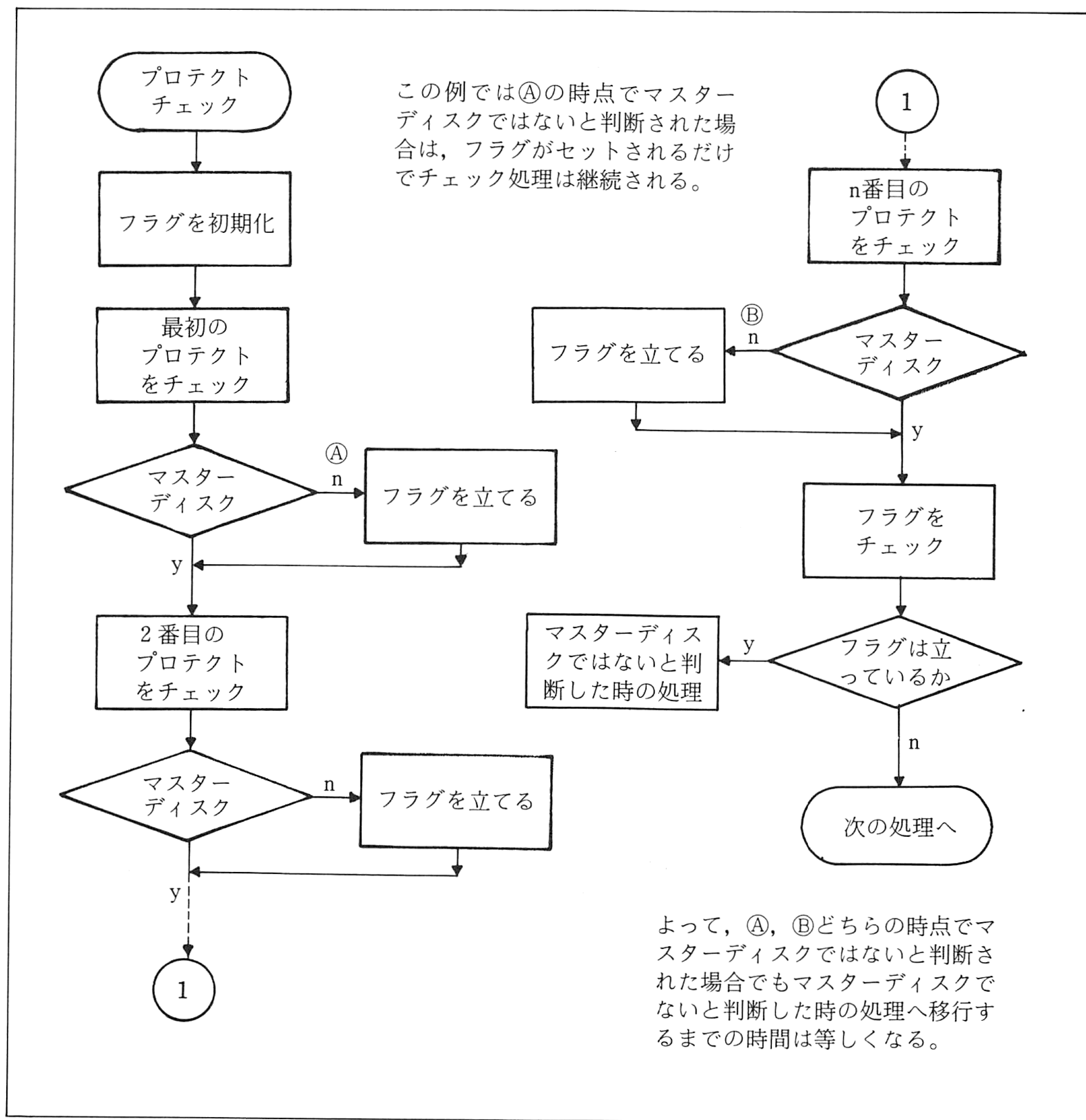
などである。

一枚のディスケット上にいくつかのプロテクトがかけられており，これをチェックして行く場合，前述のように，BOOT後，いくつかの節でチェックする場合は，さほど問題がないが，1ケ所ですべてのプロテクトをチェックする場合問題が生ずる。

これは，チェック後，マスターディスケットではないと判断された時の処理と関係があるが，**チャート 5-1** のような処理を行なうと，プロテクトのチェック処理時間の違いから，どの時点までのプロテクトがはずされたのか分ってしまう。



そこで，**チャート 5-2** のように，チェック時にマスターディスケットでないと判断した場合も，フラグをセットするのみで，すべてのプロテクトのチェックを行なった後，マスターディスケットではないと判断された場合の処理を行なえば，見かけ上，いくつかのプロテクトが外された場合も，またプロテクトが外されていない場合でも，処理時間は一定となるため，どの時点までのプロテクトが外れたのかは知ることは不可能となる。



これと異なった方法もいくつか考えられるが，重要な点は，プロテクトがどこまで外されたのかを外部から見て判断不可能なようにロジックを組むことである。

次に，この次の階段の処理，つまりマスターディスケットではないと判断した場合の処理について考案する。

この場合，一般的に次の3つの処理が考えられる。

1　その時点で処理をストップする（デッドループに入る）．

2　再度BOOTからスタートさせる。

3　何らかの対応処理を行なう。

1は，マスターディスケットではないと判断された時点でデッドループに入るタイプである。

当然，ハード的にアボート（リセットスタート）する他再起動は行えない。

この場合注意しなければならないのは，デッドループに入った時点で，ホストマシンのメインメモリー中にプロテクトのチェックルーチンなど残っていると，これをもとにプロテクトが解析される恐れがある。

よって，これらのルーチンは，デッドループに入る前にすべて消却してからデッドループに入るようにするのが好ましい。

2は，再度BOOT処理，つまりIPLロードから処理を再スタートするもので，メディアやマシンの状況によって不安定に起動が行なわれた場合，マスターディスケットであっても，そうでないと判断される場合などに有効である。

ただし，プロテクト自体が，このような状況に左右されるものである場合は問題であり，プロテクトの再考を用する。

この方法だと，マスターディスケットであると判断されなければ何回でも同じ動作をくり返すので，何回目のトライアルであるかをカウントし，ある回数に達したら，1の処理などに移行するようにすべきである。

3の方法はさらにいくつかのバリエーションが考えられる。

たとえば，マスターディスクではないことを表示する，ディスケットをフォーマッティングしてしまい，すべてデータを消してしまう，他のソフトを起動させるなどが考えられるが，中にはマスターディスケットではないと判断した場合でもある一定期間（たとえば，使用回数，使用通算時間などをカウントする）は正常に動作するが，それ以上になると，プログラムを消してしまう，と言った処理も考えられる。

この方法は，コピー品に手を焼く米国の大手ソフトメーカーが実際に使用したことがある。

いずれにせよ，マスターディスクではないと判断した場合，そのディスク自体を再起動不能（またはそのディスク自体に何らかの書き込み動作を行なう）にするような処理は，よほどプロテクトのチェック処理がしっかりしていないと，誤ってマスターディスクを使用不能

にする可能性があり，即ユーザークレームにつながる可能性が大きく，問題が多い。

よって，よほど慎重に扱わなければならないことに注意していただきたい。

さて，どんなプロテクトの強力なソフトであっても，一旦起動すると「無防備」である場合が多い。

つまり，起動後何らかの方法でプログラムがアボートされて，プログラムやプロテクトを解析されてしまう場合が良くある。

これらはプロテクトと直接は関係がないが，プロテクトを最大限生かすには考慮しなければならない点である。

たとえば，起動後，ディスケットが抜かれてしまったため，リード動作が行なえずアボートして，コマンドレベルに戻ってしまった，プロテクトを兼たオリジナルのDOSを使用していたが，メインプログラムのロード中にディスケットが抜かれたため，アボートし，DOSのコマンドレベルに戻ってしまったため，プログラム，プロテクトはおろか，DOSまでも解析されてしまったなどである。

このような場合の対策としては，ディスクのリード，ライトで本来エラーが発生しないはずのトラック，セクターでエラーが発生した場合は，正常にリード，ライトが行なえるまで，無制限のリトライを行なうか，または，エラーが発生した場合でも直接コマンドレベルに戻らず，メモリー上のプログラムをすべて消してから戻るようにするか，BOOTのルーチンへ飛ばすなどの処理が必要である。

ホストマシン内の標準リード，ライトルーチンを使用すると，エラー時に無条件でコマンドレベルに戻るようになっている場合が多いので，これは使用せず，独自に作成したほうが良い。

また，リセットがかけられた場合は，メインメモリー上にプログラムが残されたまま再起動されるような位置には置かず，リセットスタート時にイニシャライズされるようなエリアに重要なルーチンを置くと良い。

データなどは，単独ではまず解析されることはないので，どこに置いても良い。

ただし，ある種のゲームソフトのようにデータそのものが重要な意味を持つ場合などは，そのままの形では置かず，エンコード，または暗号化して置くようにする。

一部のソフトで使用されるパスワードなども当然このような処理が必要である。

5-2 プロテクトとアフターケア

プロテクトを持つソフトウェアは，商品がブラックボックス化する点で何かと批判が多い。

このため，プロテクトを持つソフトを商品として見た場合，ソフトウェアメーカー，またはプログラム開発者が常に考えなければならないのがアフターケアの問題である。

ソフトウェアを，ユーザーが好みに応じて変更できない場合は，そのソフトウェアの操作性，バグなどのクレームがより強い形でユーザーから出される。

この対策として，定期的なバージョンアップ，メインテナンスなどが必要となる。

よって，プロテクトをかけたソフトを商品として売る場合，メインテナンスなどの経費とプロテクトをかけないで，ユーザーが自由に変更できるようにした場合（当然コピーが行なわれる可能性が大きい）の第三者のコピーによる損益のバランスが取れないようなものであれば，プロテクトをかけない方が良い場合がある。

また，アフターケアやメインテナンスがしっかりしているソフトメーカーの商品は，アフターケアやメインテナンスそのものが大きな商品価値となるので，コピーによる損害は少ないと言われるほどで，いかにこの点が重要であるかが分る。

ソフトウェアの複製は，他のメディアの複製と異なり，まったく機能上同一のものを生み出すためプロテクトが必要となるわけであるが，どんなソフトウェアでもプロテクトをかければ，コピーによる損害が減り，利益率が良くなると言った考えは誤りであることを良く認識してプロテクトを利用していただきたい。

付録 1

MB8876A MB8877A仕様書

本書中で8876と称しているのはMB8876A，MB8877Aの両方どちらにも適応される。

富士通集積回路　マイクロコンピュータファミリー
8 ビット16ビット編
（1982年 4 月版） 富士通刊　より抜粋

FLOPPY DISK FORMATTER /CONTROLLER

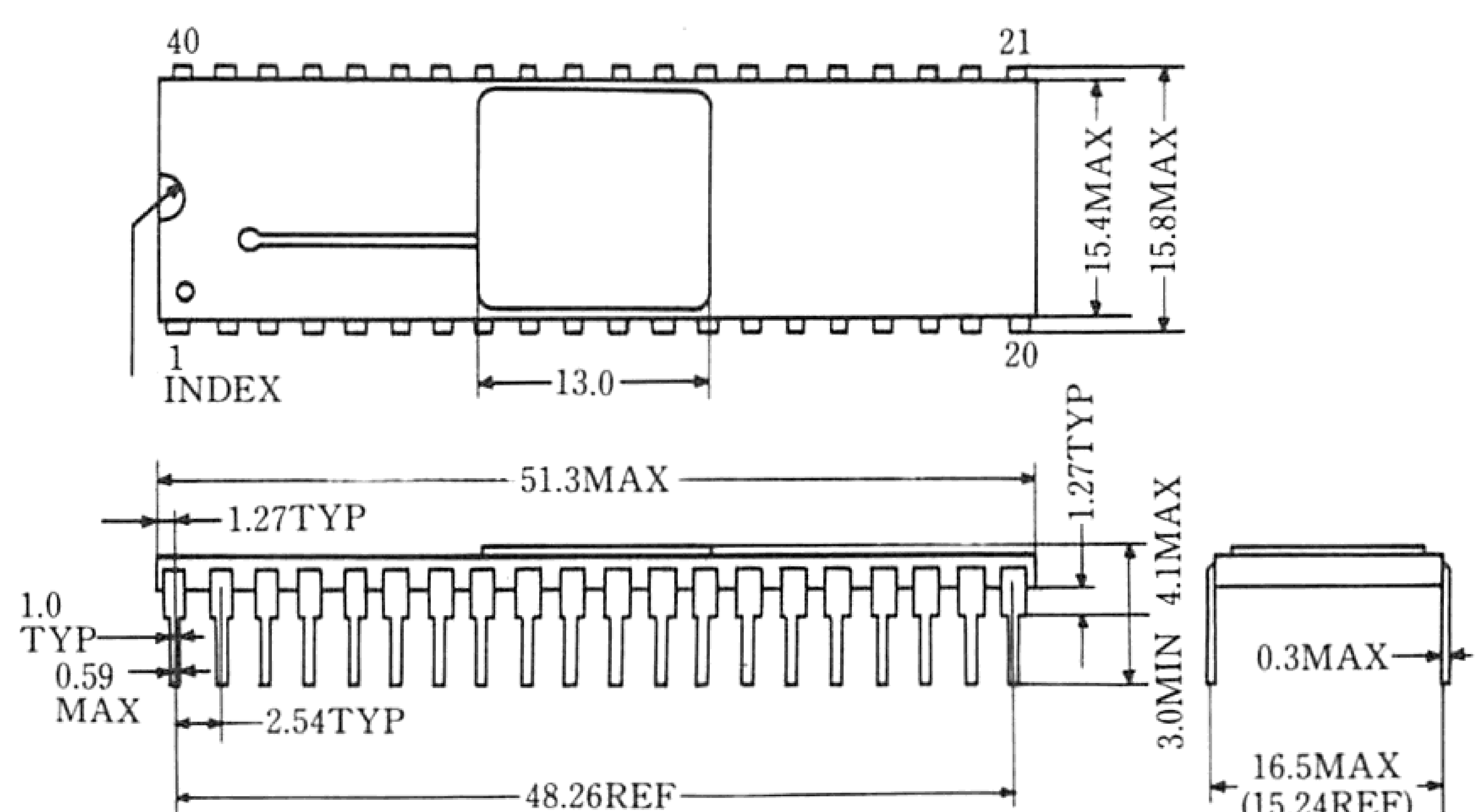
MB8876A/MB8877A は，フロッピイディスク・フォーマッタ／コントローラの機能を有するワンチップ LSI です。本 LSI は，フロッピイディスクドライブ装置とプロセッサとの間に介在し，ディスクに対してデータの書き込み，読み出し，並びにメカニカル部分の駆動や制御信号の検出を行ないます。

片面フロッピイディスクから，倍密度フロッピイディスク，ミニフロッピイディスクに適用できます。

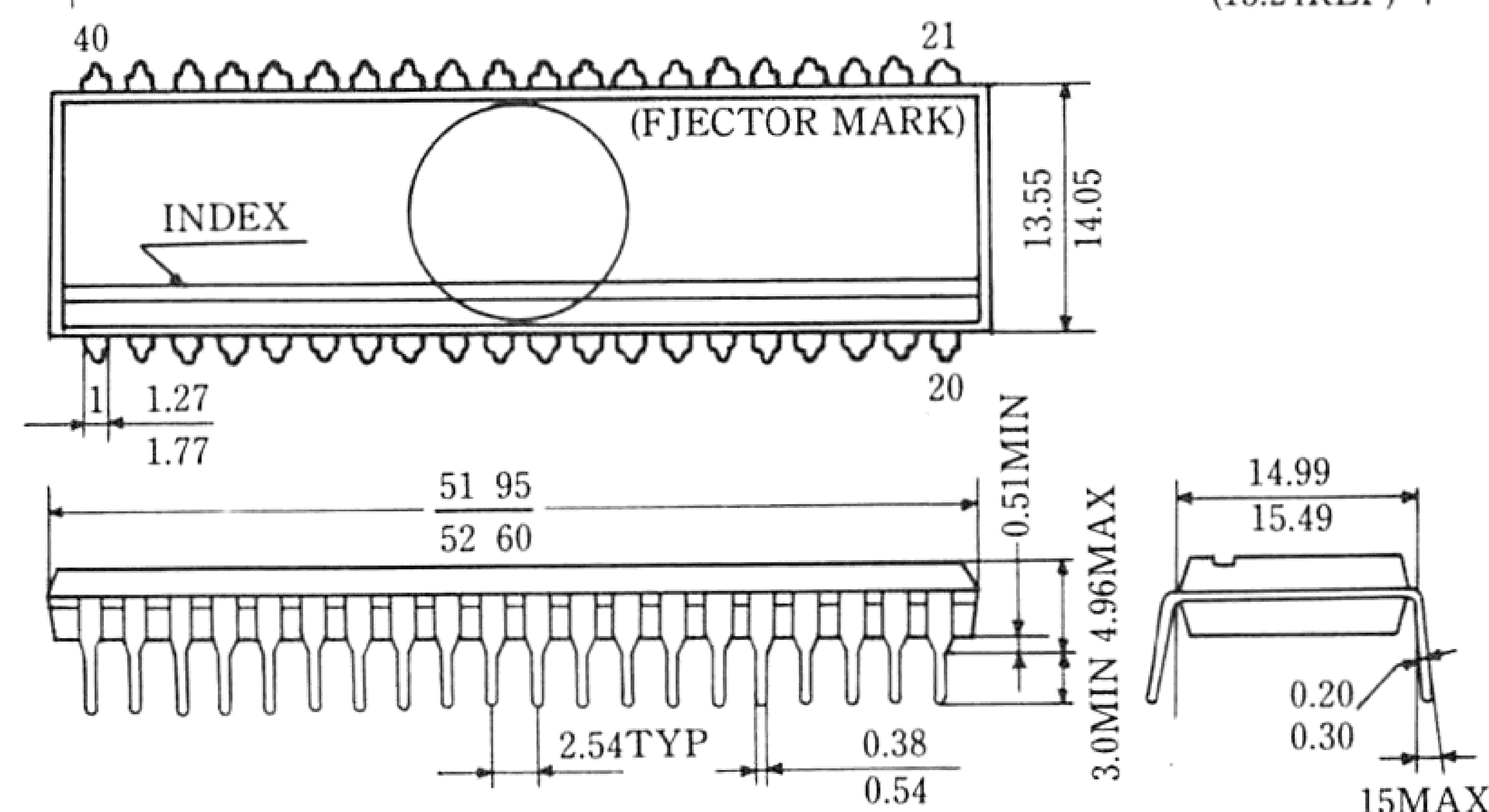
シングルデンシティ（FM 変調方式）では，IBM3740 フォーマット，グブルデンシティ（MFM 変調方式）では，IBM システム34フォーマットに従って使用されます。

外形寸法図
（単位：mm）

セラミックパッケージ



プラスチックパッケージ



特　　長

・IBM ソフトセクタフォーマットに互換性有
・トラックシーク後の自動ベリファイ動作可能
・シングルデンシティ，ダブルデンシティ両方に適用可能
　　シングルデンシティ　　　IBM3740 フォーマット
　　ダブルデンシティ　　　　IBM システム34フォーマット
・単一／連続のセクタリード，セクタライトが可能
・両面ディスクのサイド比較機能内蔵
・トラックライト，トラックリード，イニシャライゼーション可能
・ヘッドセトル，ヘッドエンゲージ時間の変更可能
・ステップ出力，ディレクション出力によるポジショニング
・データ入出力はダブルバッファ
・データ転送は，DMA 転送及びプログラム転送が可能
・全ての入出力端子は完全 TTL コンパチブル
・ライトプリコンペンセーション機能内蔵（FM，MFM）
・＋５Ｖ単一電源動作
・FD1791-02 及び FD1793-02 にピン配列，機能で互換性有
・MB8876A は負論理データバス，MB8877A は正論理データバス

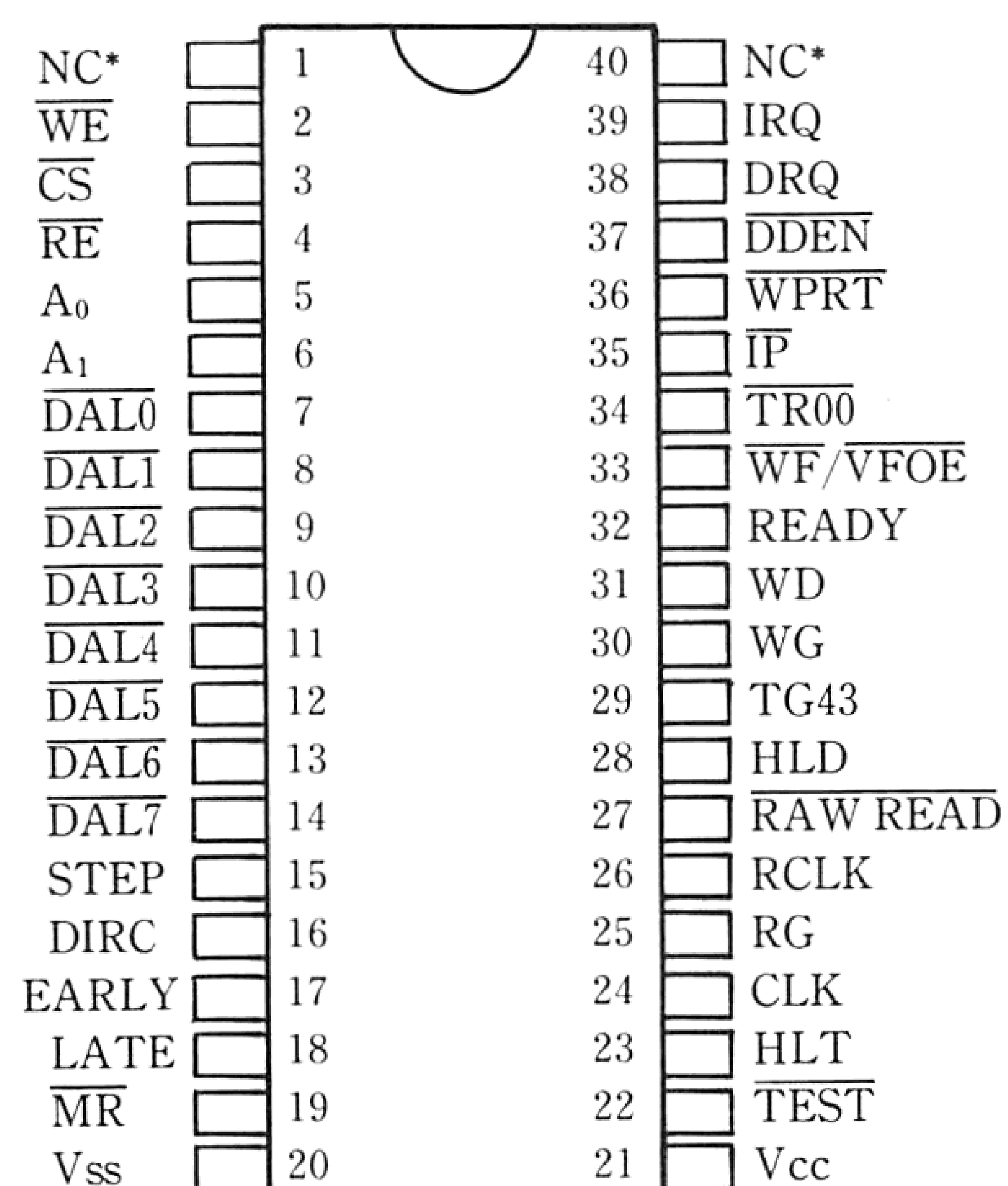
最大定格

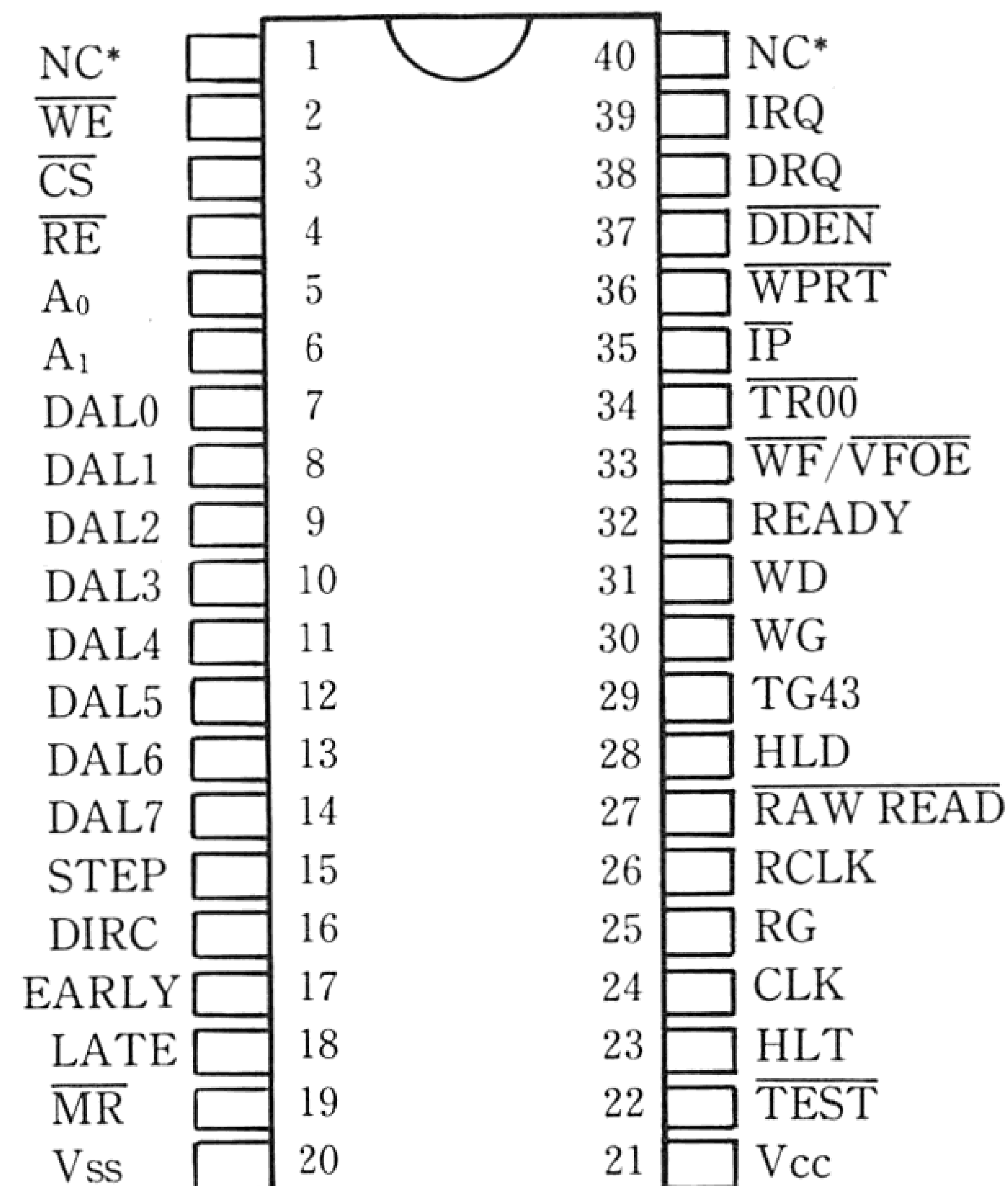
項　　目	記　号	定　　格	単　位
電　源　電　圧	V_{CC}	－0.3～＋7.0	V
	V_{SS}	0	V
入　力　電　圧	V_{IN}	－0.3～＋7.0	V
出　力　電　圧	V_{OUT}	－0.3～＋7.0	V
動　作　温　度	T_A	0～70	℃
保　存　温　度	T_{stg}	－55～＋150	℃

推奨動作条件

項　　目	記　号	定　　格	単　位
電　源　電　圧	V_{CC}	＋5.0±5％	V
	V_{SS}	0	V
動　作　温　度	T_A	0～70	℃

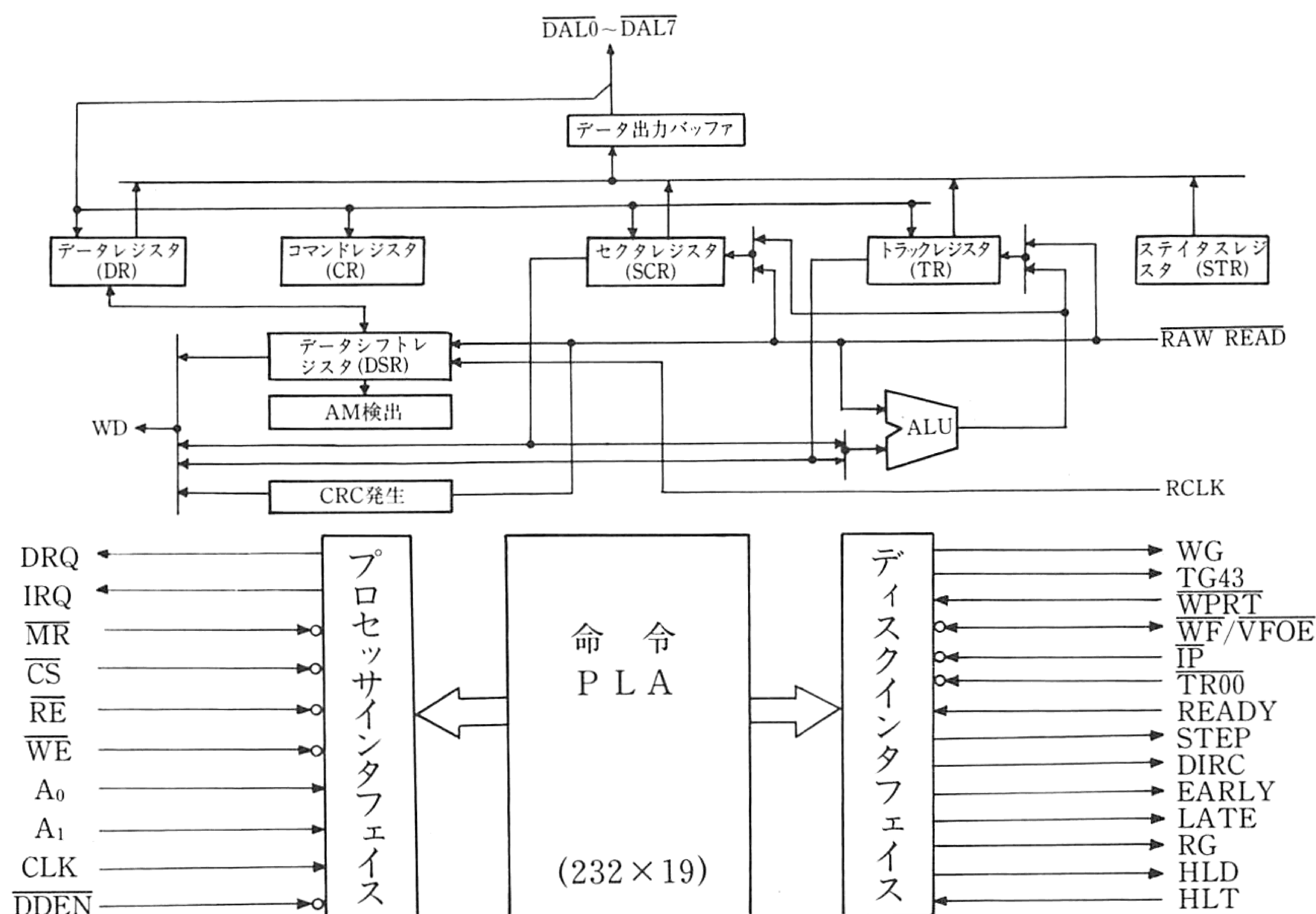
端子配列図（TOPVIEW）





NC*: No internal connection

ブロックダイヤグラム



電気的特性

1.直流特性

(V_{CC}＝＋5V±5%，V_{SS}＝0V，T_A＝0℃～70℃)

項目	記号	条件	規格値 最小	標準	最大	単位
入力電圧"H"レベル	V_{IH}		V_{SS}＋2.0	—	V_{CC}	V
入力電圧"L"レベル	V_{IL}		V_{SS}－0.3	—	V_{SS}＋0.8	V
出力電圧"H"レベル	V_{OH}	I_{OH}＝－200μA	V_{SS}＋2.4	—	—	V
出力電圧"L"レベル	V_{OL}	I_{OL}＝1.8mA	—	—	V_{SS}＋0.4	V
スリーステート入力電流（オフ状態）	I_{TSI}	V_{IN}＝0.4～2.4V	—	—	10	μA
入力リーク電流 HLT，$\overline{\text{TEST}}$，$\overline{\text{WPRT}}$ $\overline{\text{DDEN}}$は除く	I_{IN1}	V_{IN}＝0～5.25V	—	—	2.5	μA
入力リーク電流 HLT，$\overline{\text{TEST}}$，$\overline{\text{WPRT}}$，$\overline{\text{DDEN}}$	I_{IN2}	V_{IN}＝0～5.25V	—	—	100	μA
出力リーク電流	I_{LOH}	V_{OH}＝2.4V（オフ）	—	—	10	μA
消費電力	P_d		—	—	350	mW

2.動作特性

(1)リードタイミング　（図1，図3参照）

(V_{CC}＝＋5.0±5%，V_{SS}＝0V，T_A＝0～70℃)

項目	記号	条件	規格値 最小	標準	最大	単位
アドレスセットアップタイム（$\overline{\text{RE}}$↓まで）	t_{SET}		50	—	—	ns
アドレスホールドタイム（$\overline{\text{RE}}$↑から）	t_{HLD}		10	—	—	ns
$\overline{\text{RE}}$パルス幅	t_{RE}		280	—	—	ns
DQRリリースタイム（$\overline{\text{RE}}$↓からDRQ↓）	t_{DRR}		—	—	250	ns
IRQリリースタイム（$\overline{\text{RE}}$↓からIRQ↓まで）	t_{IRR}		—	—	500	ns
データ遅延時間（$\overline{\text{RE}}$↓から）	t_{DACC}	C_L＝25pf	—	—	250	ns
データホールドタイム（$\overline{\text{RE}}$↑から）	t_{DOH}	C_L＝25pf	50	—	150	ns
DRQサービスタイム	t_{SEVR}	t_C＝2μs	—	—	13.5	μs

(2)ライトタイミング　(図2，図3参照)

項目	記号	条件	規格値			単位
			最小	標準	最大	
アドレスセットアップタイム ($\overline{\mathrm{WE}}$↓まで)	t_{SET}		50	—	—	ns
アドレスホールドタイム ($\overline{\mathrm{WE}}$↑から)	t_{HLD}		10	—	—	ns
$\overline{\mathrm{WE}}$パルス幅	t_{WE}		200	—	—	ns
DRQリリースタイム ($\overline{\mathrm{WE}}$↓からDRQ↓)	t_{DRR}		—	—	250	ns
IRQリリースタイム ($\overline{\mathrm{WE}}$↓からIRQ↓)	t_{IRR}		—	—	500	ns
データセットアップタイム ($\overline{\mathrm{WE}}$↑まで)	t_{DS}		250	—	—	ns
データホールドタイム ($\overline{\mathrm{WE}}$↑から)	t_{DH}		0	—	—	ns
DRQサービスタイム	t_{SEVW}*	$\overline{\mathrm{DDEN}}$＝“L”	—	—	11.5	μs

＊ CLK＝1MHz＝時，値は2倍になる。

(3)各種タイミング　(図4参照)

項目	記号	条件	規格値			単位
			最小	標準	最大	
CLK“L”レベル幅	t_{CD1}		230	—	20000	ns
CLK“H”レベル幅	t_{CD2}		200	—	20000	ns
STEPパルス幅	t_{STP}*	$\overline{\mathrm{DDEN}}$＝“L” $\overline{\mathrm{DDEN}}$＝“H”	2 4	— —	— —	μs
DIRCセットアップタイム (STEP↑まで)	t_{DIR}*		12	—	—	μs
$\overline{\mathrm{MR}}$パルス幅	t_{MR}*		50	—	—	μs
$\overline{\mathrm{IP}}$パルス幅	t_{IP}*		10	—	—	μs
$\overline{\mathrm{WF}}$パルス幅	t_{WF}*		10	—	—	μs

＊ CLK＝1MHzの時，各々の値は2倍になる。

(4)入力データタイミング　(図5参照)

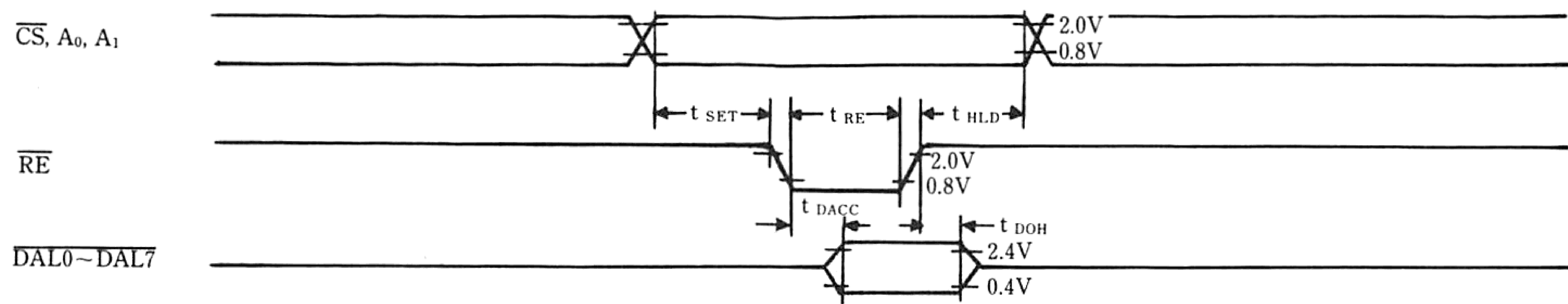
項目	記号	条件	規格値			単位
			最小	標準	最大	
$\overline{\text{RAW READ}}$パルス幅	t_{pW}*		100	—	250	ns
RCLK変化から $\overline{\text{RAW READ}}$の立ち下りまで	t_d		40	—	—	ns
$\overline{\text{RAW READ}}$の立ち下りから RCLKの変化まで	t_{cd}	$\overline{\text{DDEN}}$＝"L"	40	—	—	ns
$\overline{\text{RAW READ}}$の立ち上りから RCLKの変化まで	t_{cs}	$\overline{\text{DDEN}}$＝"H"	40	—	—	ns
$\overline{\text{RAW READ}}$サイクルタイム	t_{bc}	$\overline{\text{DDEN}}$＝"L" $\overline{\text{DDEN}}$＝"H"	—	2*, 3*, 4* 2*, 4*	—	μs
RCLK "H" レベル幅	t_a	$\overline{\text{DDEN}}$＝"L" $\overline{\text{DDEN}}$＝"H"	0.8 0.8	1 * 2 *	20 20	μs
RCLK "L" レベル幅	t_b	$\overline{\text{DDEN}}$＝"L" $\overline{\text{DDEN}}$＝"H"	0.8 0.8	1 * 2 *	20 20	μs
RCLKサイクルタイム	t_c	$\overline{\text{DDEN}}$＝"L" $\overline{\text{DDEN}}$＝"H"	—	2 * 4 *	—	μs

＊ CLK＝1MHzの時，各々の値は2倍になる。

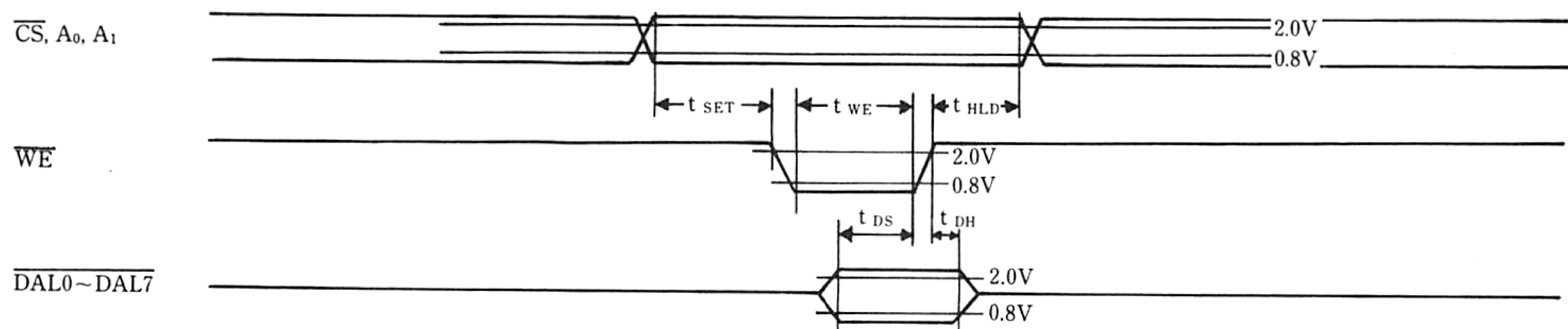
(5)出力データタイミング　(図6参照)

項目	記号	条件	規格値			単位
			最小	標準	最大	
WDパルス幅	t_{WD}	CLK＝2MHz FM MFM	450 150	500 200	550 250	ns
WGセットアップタイム	t_{WG}	CLK＝2MHz FM MFM	— —	2 1	— —	μs
WGホールドタイム(WDから)	t_{Wf}	CLK＝2MHz FM MFM	— 1	2 —	— 2	μs
EARLY(LATE)セットアップタイム	t_S	CLK＝2MHz MFM	125	—	—	ns
EARLY(LATE)ホールドタイム(注)	t_H	CLK＝2MHz MFM	－50	—	—	ns
WDバリッド時間1	t_{D1}	CLK＝1MHz MFM CLK＝2MHz MFM	200 30	— —	— —	ns
WDバリッド時間	t_{D2}	CLK＝1MHz MFM CLK＝2MHz MFM	50 50	— —	— —	ns

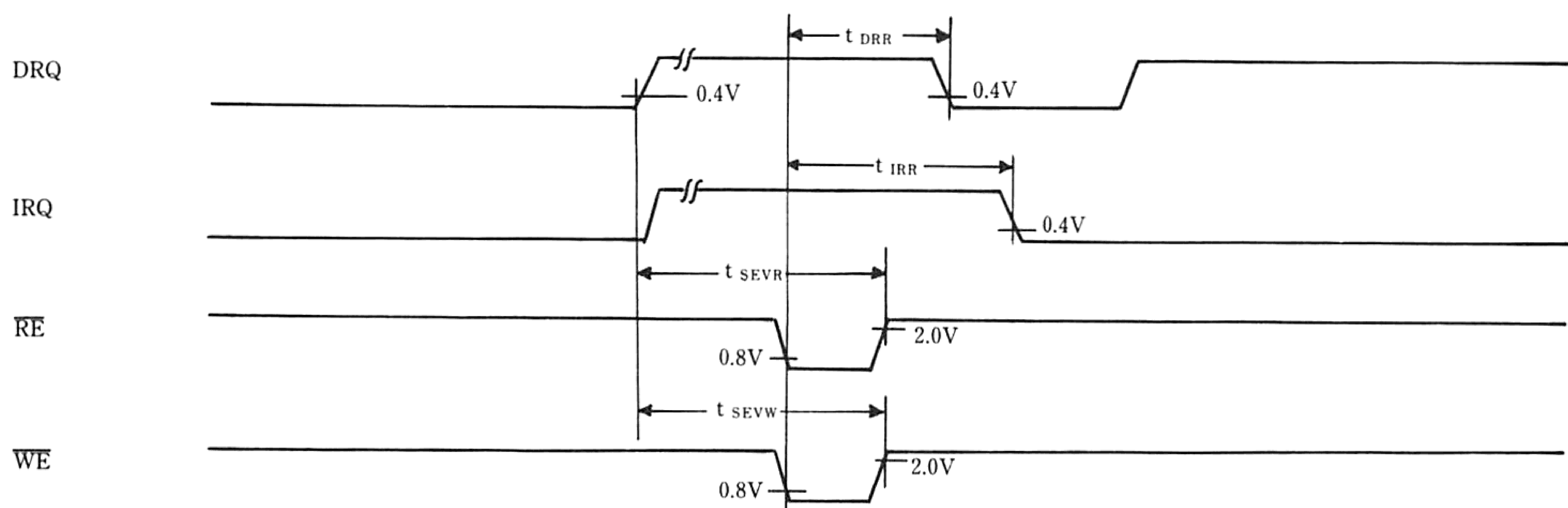
(注) 本項の値－50nsはWDの立ち下りより50nsec以前に現在のWD信号に対するEARLY(LATE)が変化することを示す。



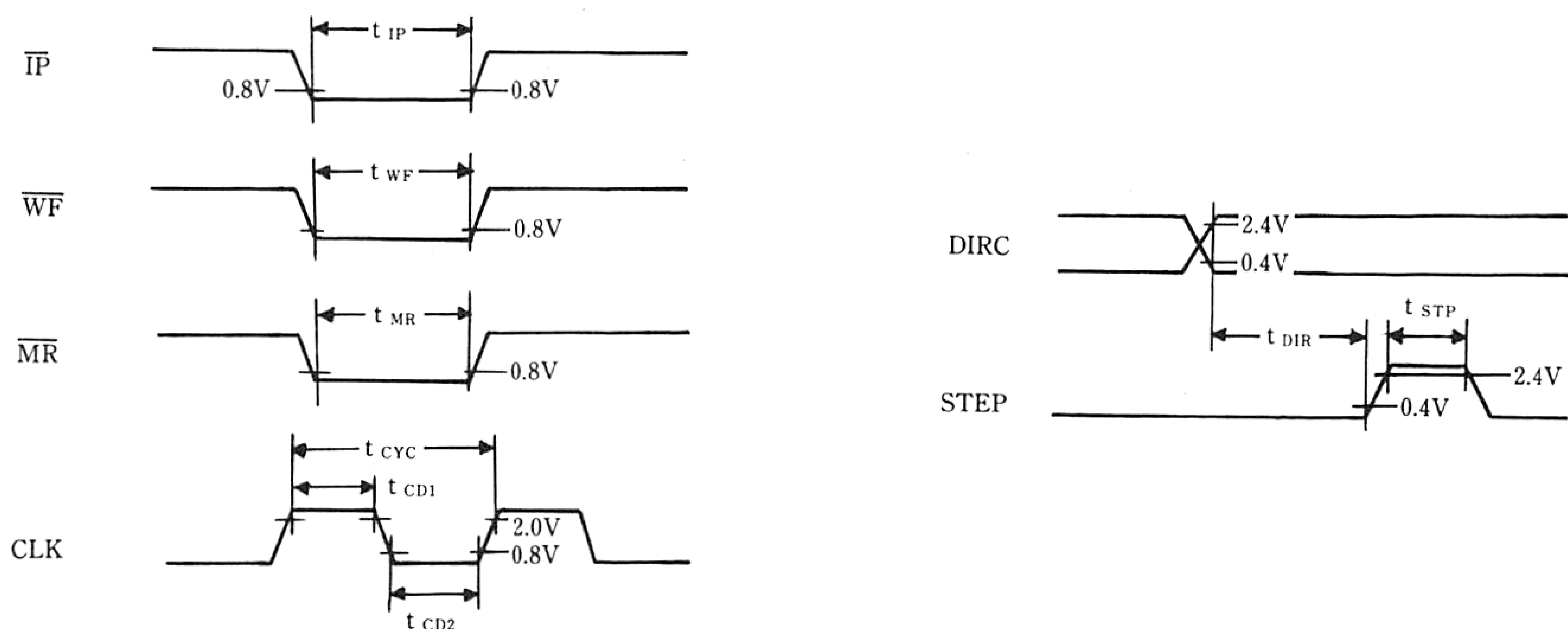
〔図１〕 リードタイミング



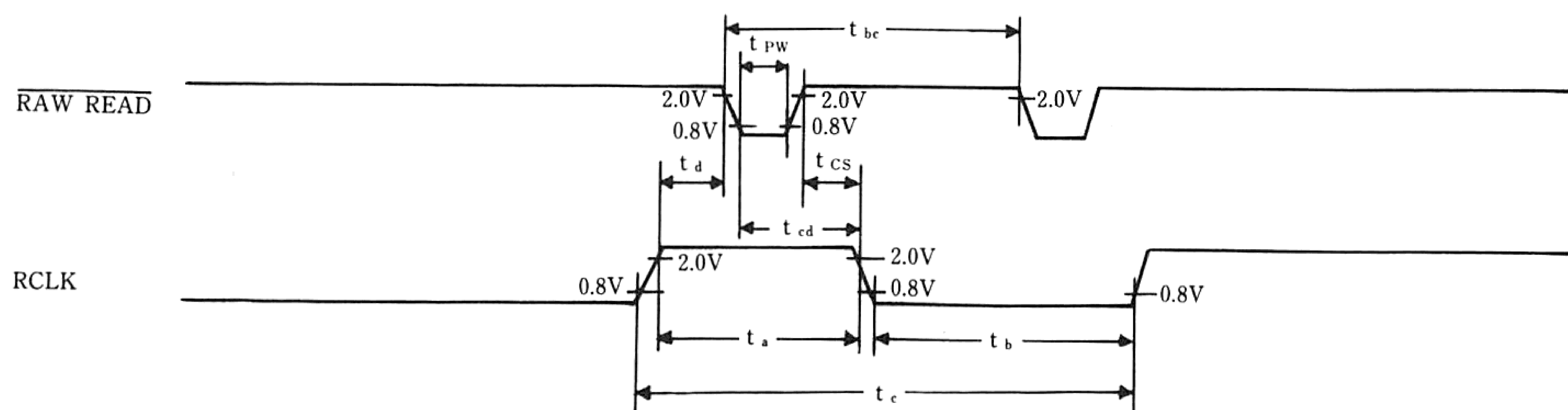
〔図２〕 ライトタイミング



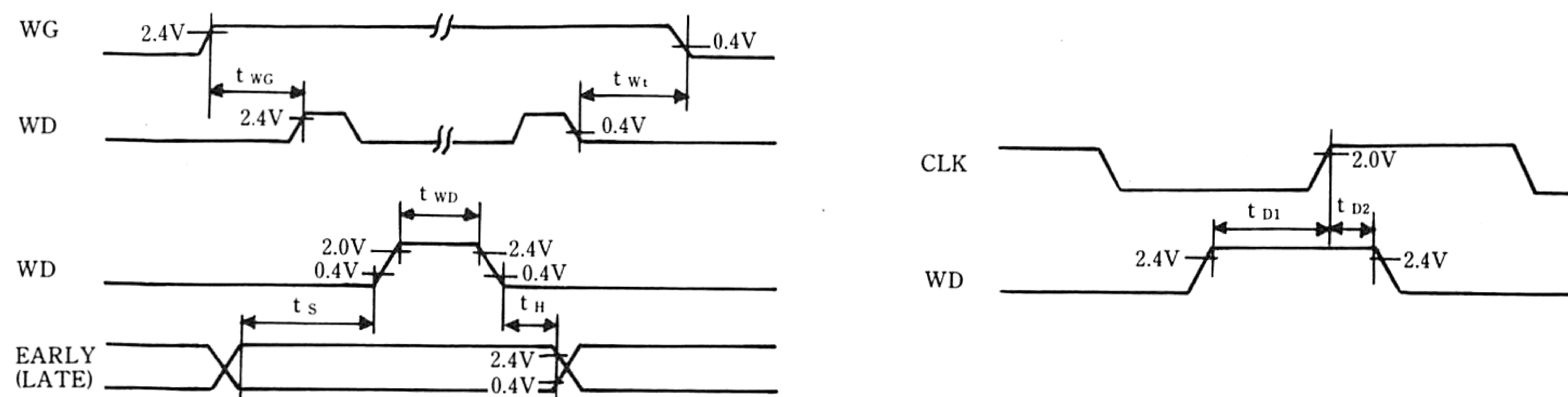
〔図３〕 DRQ,IRQリリースタイミング



〔図４〕 各種タイミング



〔図５〕 入力データタイミング



〔図６〕 出力データタイミング

端子機能説明

端子番号	端子名称	記号	I/O	機能説明
20	POWER SUPPLIES	V_{SS}	I	接地端子。
21		V_{CC}	I	＋５Ｖ電源端子。
19	MASTER RESET	$\overline{MR}$	I	$\overline{MR}$＝“L”とする事で全ての動作を停止します。 $\overline{MR}$の立ち上りでリストアコマンドを実行します。
2	WRITE ENABLE	$\overline{WE}$	I	内部レジスタへの書き込みストローブ信号入力端子。 $\overline{CS}$＝“L”の時，A_1，A_0で指定される内部レジスタへ，$\overline{DAL}$(DAL)の内容が書き込まれます。
3	CHIP SELECT	$\overline{CS}$	I	チップ選択信号であり$\overline{CS}$＝“L”の時，プロセッサとのデータ転送が可能となります。
4	READ ENABLE	$\overline{RE}$	I	内部レジスタから読み出しストローブ信号入出力端子。 $\overline{CS}$＝“L”の時，A_1，A_0で指定される内部レジスタの内容が$\overline{DAL}$(DAL)に出力されます。
5	REGISTER SELECT LINES	A_0	I	内部レジスタ選択端子であり，A_1，A_0により，ＣＲ，STR，TR，SCR，DRのいずれかが選択されます。
6		A_1		
7 ～ 14	DATA ACCESS LINES	$\overline{DAL0}$ DAL0 ～ $\overline{DAL7}$ DAL7	I/O	８ビットの双方向性スリーステートの入出力端子。 $\overline{CS}$＝“H”の時，この端子はハイインピーダンス状態となります。 MB8876Aでは負論理，MB8877Aでは正論理。
24	CLOCK	CLK	I	基本クロック入力端子。標準サイズのフロッピイディスクでは2MHz，ミニフロッピイディスクでは1MHzのクロックを入力してください。
38	DATA REQUEST	DRQ	O	リード動作の時，ディスクから読み出されたデータが，ＤＲに用意されたことを示し，ライト動作の時，ディスクへ書き込むデータを必要とする事を示します。この出力はオープンドレイン出力であり，外部で10kΩ程度の抵抗でプルアップする必要があります。
39	INTERRUPT REQUEST	IRQ	O	コマンドの終了，打ち切り，及びタイプⅣコマンドの指定に従って発生する出力信号。この出力はオープンドレイン出力であり，外部で10kΩ程度の抵抗でプルアップする必要があります。
15	STEP	STEP	O	ヘッドを移動させるための出力信号であり，１トラック移動のために１つのパルスを出力します。
16	DIRECTION	DIRC	O	ヘッドの移動方向を指定する出力信号であり，“L”レベルで外側へ，“H”レベルで内側へ移動する事を示します。
17	EARLY	EARLY	O	ディスクへ書き込むデータ(ＷＤ)を早い方へシフトさせるべきである事を示す，ライトプリコンペンセイション用端子。
18	LATE	LATE	O	ディスクへ書き込むデータ(ＷＤ)を遅い方へシフトさせるべきである事を示す，ライトプリコンペンセイション用端子。
22	TEST	$\overline{TEST}$	I	ＬＳＩのテスト時に用いられる端子であり，通常はオープンか，プルアップしてください。

端子番号	端子名称	記号	I/O	機能説明
23	HEAD LOAD TIMING	HTL	I	ヘッドロード指定（HLD出力）後の応答入力信号であり，ヘッドエンゲージ状態（安定してヘッドロードが行なわれた状態）の時“H”レベルとしてください。
25	READ GATE	RG	O	RG＝“H”でSYNCバイトを検出した事を示します。
26	READ CLOCK	RCLK	I	ディスクから読み出されるシリアルデータにかけるデータウィンドウ信号であり，シリアルデータと共に入力してください。
27	RAW READ	$\overline{\text{RAW READ}}$	I	ディスクから読み出されるシリアルデータそのものであり，クロックビット，データビットがシリアルに混在したものを入力してください。
28	HEAD LOAD	HLD	O	ヘッドロード(メディアに押しつける事)を示す信号であり，“H” レベルでこれを示します。
29	TRACK GREATER THAN 43	TG43	O	TG43＝“H”でヘッドがトラック44以上にある事を示します。 TG43＝“L”でヘッドがトラック00～43にある事を示します。
30	WRITE GATE	WG	O	WG＝“H”でディスクへデータを書き込んでいる事を示します。
31	WRITE DATE	WD	O	ディスクへの書き込みデータパルスを出力する端子です。シングルデンシティで500ns，ダブルデンシティ250nsのパルスを出力します（1MHz時は各々２倍）。
32	READY	READY	I	READY＝“H”でディスクドライブが動作可能状態である事を知り，タイプＩコマンド以外，この入力が“H”レベルの時，コマンドを実行します。
33	WRITE FAULT / VFO ENABLE	$\overline{\text{WF/VFOE}}$	I/O	WG＝“H”で，書き込み時の障害検出入力として動作し，WG＝“L”の時，ＬＳＩがディスクからデータを読んでいる事を“L”レベルを出力して通知します。本端子はオープンドレイン形式の入出力端子であり，入力モード，出力モードはWGのレベルにより決まります。
34	TRACK 00	$\overline{\text{TR00}}$	I	ディスクのヘッドがトラック０の位置にあることを検出する入力信号であり，“L”レベルでこれを検出します。
35	INDEX PULSE	$\overline{\text{IP}}$	I	ディスクの基準位置(インデックスホール)を検出する入力端子である。ディスクが１回転する毎に１つのパルスをＬＳＩに加えてください。
36	WRITE PROTECT	$\overline{\text{WPRT}}$	I	ディスクの書き込み禁止を検出する入力端子。本信号が“L”レベルの時，書き込みは行ないません。
37	DOUBLE DENSITY	$\overline{\text{DDEN}}$	I	シングルデンシティ動作，ダブルデンシティ動作を指定する入力信号端子。 $\overline{\text{DDEN}}$＝“H”でシングルデンシティ。 $\overline{\text{DDEN}}$＝“L”でダブルデンシティ。

レジスタ説明

1．コマンドレジスタ（CR）

書き込み専用の8ビットレジスタです。このレジスタには，FDCに実行させる動作に応じたコマンドを，プロセッサ側から書き込みます。コマンドの書き込み動作は，フォースインタラプトコマンドを除き，FDCが前コマンド動作の終了後でなければなりません。

2．ステイタスレジスタ（STR）

読み出し専用の8ビットレジスタです。このレジスタは，FDCの内部状態，コマンド実行における処理状態，ディスクドライバの状態を示します。各ビットの意味は，実行中及び実行完了したコマンドにより変化します。

3．データレジスタ（DR）

書き込み，読み出し可能の8ビットレジスタです。ディスクリード時には，ディスクから読み出されたデータがロードされ，ディスクライト時には，このレジスタに書き込まれたデータがディスクに書き込まれます。

シーク動作時には，目的のトラック番号を書き込みます。

4．データシフトレジスタ（DSR）

このレジスタは，外部からアクセスすることはできない8ビットのシフトレジスタです。

ディスクライト時には，DRからDSRへ並列転送されたデータが，ライトデータとして，FMもしくはMFMで変調され，WD端子から直列に出力されます。

ディスクリード時には，$\overline{\text{RAW READ}}$端子から入力するデータが直列にDSRに入り，復調されて1バイト単位でDRへ並列転送されます。

5．トラックレジスタ（TR）

書き込み，読み出し可能な8ビットレジスタです。このレジスタは，マスタリセット(ML＝“L”から“H”の立ち上り）によって（FF)$_H$から順次減少し，$\overline{\text{TR00}}$が“L”レベルになった時に（００)$_H$となります。

このレジスタは通常ディスクのリード／ライト用ヘッドのあるトラックの番号がセットされます。FDCはコマンドにより，この値を更新することも更新しない事も可能です。

リードデータ，ライトデータコマンドでは，この内容とディスクから読み出されたIDフィールドのトラック番号を比較して，一致したセクタに対しリード／ライトを行ないます。

6．セクタレジスタ（SCR）

書き込み，読み出し可能な8ビットシフトレジスタです。リードデータ，ライトデータコマンドでは，このレジスタとディスクから読み出されたIDフィールドのセクタ番号を比較し，一致したセクタに対してリード／ライトを行ないます。

リードアドレスコマンドでは，ID フィールドのトラック番号が保持されます。

7．CRC 回路

CRC（Cyclic Redundancy Check）回路は，ディスクとの間で転送されるシリアルデータに対し，書き込み，読み出しデータの誤りがあるかをチェックする為の回路です。

ディスクライト動作では，書き込みデータから自動的に CRC コードを作成しディスクへ書き込みます。ディスクリード動作では読み出されたシリアルデータから CRC コードを作成し，ディスク内の CRC と比較しチェックします。

CRC 生成多項式は次の通りです。

$$G\ (X) = X^{16} + X^{12} + X^{5} + 1$$

8．演算回路（ALU）

演算回路は，シリアルデータの比較，インクリメント（＋1）デクレメント（－1），スルー（±0）の機能を有し，レジスタの内容の更新，比較，照合のために使用されます。

9．アドレスマーク検出（AM 検出）

アドレスマーク検出回路は，ディスクから読み出されたシリアルビットセルから特定のビットパターンを有するデータを検出力する回路です。検出されるデータは，インデックスマーク，ID アドレスマーク，データマーク，デリーテッドデータマークです。

10．データ変調

プロセッサとディスク間に転送されるデータを，シングルデンシティでは FM 方式，ダブルデンシティでは MFM 方式で変調及び復調します。

11．命令 PLA

FDC はマイクロプログラムにて制御されます。命令 PLA（Programable Logic Array）はこのマイクロプログラムが格納されており，FDC の各部に制御信号を発します。このマイクロプログラムの大きさは約232×19（ビット）です。

動作

FDC は，プロセッサ側のインタフェイスと，フロッピイディスク側のインタフェイスを持っています。プロセッサインタフェイスは，データ転送のための信号線とその制御線からなります。フロッピイディスクインタフェイスは，ディスクとのデータ転送用の信号線とその制御線，機械的部分の駆動用信号線とその制御線，フロッピイディスクの状態検出用線からなります。

プロセッサインタフェイス

プロセッサインタフェイスは，FDC 内部のレジスタと，プロセッサのデータ転送を行なう

ためのインタフェイスです。プロセッサはFDCに必要に応じたデータ，コマンドなどをこのインタフェイスを介して，内部レジスタと転送を行ないます。

このインタフェイスの信号線には，データ線と，レジスタ選択線，リード線，ライト線，チップセレクト線があります。プロセッサは，チップセレクト線，レジスタ選択線，データ線に所要のレベルを与えて，リード線もしくはライト線を制御して，データ転送を行ないます。内部レジスタの選択を**表1**に示します。

〔表1〕レジスタ選択

$\overline{CS}$	A_1	A_0	$\overline{RE}$＝"L"	$\overline{WE}$＝"L"
"H"	X	X	NON SELECT（$\overline{DAL}$はHiインピーダンス）	
"L"	"L"	"L"	ステイタスレジスタ	コマンドレジスタ
"L"	"L"	"H"	トラックレジスタ	トラックレジスタ
"L"	"H"	"L"	セクタレジスタ	セクタレジスタ
"L"	"H"	"H"	データレジスタ	データレジスタ

また，コマンド動作終了の通知の機能を持つ割り込み要求線(IRQ)及びディスクとのデータ転送におけるデータ読み出し／書き込み要求線（DRQ）があります。これによりプロセッサのプログラムの軽減，DMA転送のタイミング信号作成の軽減を可能にしています。

フロッピイディスクインタフェイス

FDCは，フロッピイディスクインタフェイスとして，ディスクのリードライト用ヘッドの駆動用信号，ディスクからのデータ読み取り用端子，ディスクへの書き込みデータ信号，フロッピイディスクの状態検出用端子があります。

FDCは，プロセッサインタフェイスより，コマンドレジスタに書き込まれたコマンドに応じて，フロッピイディスクインタフェイスを通じて，フロッピイディスクを制御します。

(1)ヘッドの駆動

フロッピイディスクのヘッド駆動には，トラック方向の移動と，ヘッドをメディア(媒体)へ押しつける（ヘッドロード）動作があり，FDCは，独立にそれらの出力端子を持っています。

トラック方向の移動にはディレクション出力（DIRC），とステップ出力（STEP）により，移動の方向と移動トラック数を与えます。1トラック移動のために1つのSTEPパルスを出力しステッピインの方向をDIRC出力は"H"レベルで示します。ステップアウトの方向はDIRC出力を"L"レベルで示します。また，DIRC出力の更新は最初のステップパルス出力より12μsec以上先行します。

ヘッドロード動作は，ヘッドロード出力（HLD）によって指示されます。また，ヘッドロードの完了応答としてヘッドロードタイミング信号(HLT)があり，FDCは，HLDが，“H”レベルでかつ，HLTが“H”レベル時，ヘッドが安定してロードされているとみなします。即ち，この時ディスクとのリード／ライトが可能とみなします。

(2)ディスクリード動作

ディスクとのデータ転送には，シングルデンシティ（変調様式：FM）とダブルデンシティ（変調様式：MFM）の2種類が選択できます。FDCは$\overline{\text{DDEN}}$端子（Pin37）を“L”レベルとするとダブルデンシティ，“H”レベルとするとシングルデンシティで動作します。

FDCは，ディスクリードのために，ディスクからの生データ（データビットとクロックビットがシリアルに混在したパルス列：$\overline{\text{RAW READ}}$）とそれらのパルスに加えるウインドウとしてリードクロック（RCLK）を必要とします。FDCは，データの同期（バイト毎の同期)取られると，この時のRCLKのレベルと，データビット／クロックビットとしての$\overline{\text{RAW READ}}$の関係を保ちながら，読み出し動作を行ないます。

FDCは，リード動作を開始すると，$\overline{\text{VFOE}}$($\overline{\text{WF}}$と共用のオープンドレイン形式の入出力端子）出力を“L”レベルとします。リードトラックコマンド以外のリード動作は，次にバイトデータの同期をとります。これは，SYNCバイト検出後のアドレスマーク検出によって行ないます。シングルデンシティでは，2バイトの（00)$_H$を検出してRGを“H”レベルとし，その後10バイトの区間にわたりアドレスマークを検索します。アドレスマークが検出されると，その後有効なデータ(IDフィールドではIDフィールドCRCまで，データフィールドではデータフィールドのCRCまで）を読み込んだ後RGを“L”にします。アドレスマークが検出されないと，RGは“L”となり再び，2バイトの（00)$_H$を検索します。ディスクが6回転するまでにデータの同期がとられないと，リード動作は打ち切られます。ダブルデンシティでは，RGは4バイトの（00)$_H$を検出して“H”レベルになり，その後16バイトの区間にわたりアドレスマークの検索を行ないます。その他の動作はシングルデンシティの場合と同様です。

FDCがディスクからデータを読み込み，プロセッサに与えるデータがある場合，DRQ出力を“H”レベル，ステイタスレジスタのDRQステイタスビットを1とします。プロセッサは，このDRQ出力，もしくは，DRQステイタス従って，データを読み出す必要があります。DRQ出力，DRQステイタスビットはデータを読み出すと“L”レベル，0になります。もし，次のデータが受信されるまでにこの読み出し動作を怠ると，ステイタスのLOST DATAフラグがセットされます。

(3)ディスクライト動作

ディスクライト動作は，ライトゲート出力(WG)を“H”レベルとしてから，記録様式(FM

もしくはMFM）に従ってライトデータ出力（WDよりシリアルに出力します。ライトプロテクト（$\overline{WPRT}$）入力が“L”レベルの時，ライト動作は行ないません。また，WG＝“H”レベルの時，ライトフォールト入力（$\overline{WF}$）が“L”レベルになると書き込み動作は打ち切られます。

プロセッサは，FDCのデータリクエスト（DRQ）が，“H”レベルとなってから1バイト時間以内にデータを転送しなければなりません。もし，ディスクライト動作中にこの処理を怠たると，FDCは，$(00)_H$が転送されたものとし，この，$(00)_H$をディスクに書き込みます。この時，ステイタスレジスタの，LOST DATAフラグがセットされます。ただし，最初の1バイト目は，$(00)_H$は書かずに，コマンドを打ち切ります。

FDCは，ディスクライト動作のサービスとして，EARLY，LATEの書き込み補償用端子があります。EARLY＝“H”レベルの時，書き込みパルスは早めに伝送されるべきである事を示し，LATE＝“H”レベルの時は，遅れて伝送されるべきである事を示します。

また，書き込み電流切換のためにTG43出力をもっています。これは，トラックレジスタの内容が43より大きい場合に“H”レベルとなり，43以下の場合は“L”レベルとなります。

マスタリセットについて

FDCは，マスタリセット端子（MR）を“L”レベルにする事により全ての動作を停止します。また，この時内部レジスタは以下のようにプリセットされます。

(1) コマンドレジスタ　$(03)_H$
(2) ステイタスレジスタ　不確定（以前のステイタスを保持）
(3) データレジスタ　不確定
(4) トラックレジスタ　不確定
(5) セクタレジスタ　$(01)_H$

マスタリセットが解除された時（$\overline{MR}$が“L”から“H”へ立ち上った時）FDCは，リストアコマンドを実行します。この時のステップレートは，タイプ1コマンドのステップレートフラグ$r_1 r_0$が，$r_1=1$，$r_0=1$の場合と同様です。

コマンド説明

FDCには11種類のコマンドがあり，各コマンドは細部にわたる動作を決めるフラグを持ち，多くの動作モードを実現しています。

これらのコマンドは，4種に大別され，これをタイプⅠからタイプⅣとします。コマンド一覧を**表2**に示します。

〔表2〕コマンド一覧

タイプ	コマンド名称	動　　　　作
I	リストア	トラック0へ，ヘッドを移動する
	シーク	所定のトラックへ，ヘッドを移動する
	ステップ	ヘッドを1トラック移動する
	ステップイン	ヘッドを1トラック内側へ移動する
	ステップアウト	ヘッドを1トラック外側へ移動する
II	リードデータ	ディスクのデータ(データフィールド)を読む
	ライトデータ	ディスク(データフィールド)へデータを書込む
III	リードアドレス	ディスクのIDフィールドを読む
	リードトラック	ディスクの1トラック分の全データを読む
	ライトトラック	ディスクへ1トラック分の全データを書き込む
IV	フォースインタラプト	割り込みを発生させる

各コマンドは,タイプIVのコマンドを除き前のコマンドが終了した後(ステイタスのBUSY＝0の時) でなければ書き込んではなりません。

タイプIコマンド

タイプIコマンドは，ヘッドの移動とトラックの照合の機能を有します。各コマンドのコードとそれらの持つフラグを**表3**に示します。

〔表3〕タイプIコマンド

コマンド名称	コマンドレジスタビット (MSB)　7 6 5 4 3 2 1 0　(LSB)
リストア	0 0 0 0 h V r_1 r_0
シーク	0 0 0 1 h V r_1 r_0
ステップ	0 0 1 u h V r_1 r_0
ステップイン	0 1 0 u h V r_1 r_0
ステップアウト	0 1 1 u h V r_1 r_0

タイプIコマンドは，ステップレートフラグ($r_1 r_0$)，トラック照合フラグ(V)，ヘッドロードフラグ（h)，トラックレジスタ更新フラグ（u）を持っています。

ステップレートフラグ（$r_1 r_0$）は，ステップパルス出力（STEP）の出力間隔（ステップ

レート）を指定します。ステップレートは，このr_1，r_0と，FDCに与えられるクロック（CLK）の周波数，$\overline{\text{TEST}}$端子の状態によって**表4**のようになります。

〔表4〕ステップレートの変化

TEST		“H”もしくは開放		“L”	
r_1	r_0 ＼ CLK	2MHz	1MHz	2MHz	1MHz
0	0	3 ms	6 ms	Approx 200μs	Approx 400μs
0	1	6 ms	12ms		
1	0	10ms	20ms		
1	1	15ms	30ms		

ヘッドロードフラグ（h）は，コマンド実行開始時にヘッドをロードするか，ヘッドをメディアから離すかを指示します。

h＝1：コマンド実行開始時にヘッドロードします。

h＝0：コマンド実行開始時にヘッドを離します。

トラックレジスタ更新フラグは，ヘッド移動に際し，トラックレジスタを更新する事を指示します。

u＝1：トラックレジスタを更新します。

u＝0：トラックレジスタを更新しません。

トラック照合フラグ（V）は，ヘッド移動後，ディスクのトラック番号とトラックレジスタの照合を行なうかを指示します。

V＝1：トラックの照合を行ないます。

V＝0：トラックの照合は行ないません。

タイプIIコマンド

タイプIIコマンドは，ディスクのデータフィールドに対する書き込みと読み出し動作をします。対象となるセクタ番号とトラック番号はそれぞれ，セクタレジスタ，トラックレジスタに用意されていなければなりません。ディスクとのデータ転送はデータレジスタを介して行ないます。

タイプIIコマンドのコードを**表5**に示します。

タイプIIコマンドは，対象セクタを検索し，そのセクタに対してリード／ライトを行ないます。対象セクタ検索はIDフィールドを読み，トラック番号とトラックレジスタ，セクタ番号とセクタレジスタを比較し，IDフィールドのCRCをチェックして行ないます。オプションとして，サイド番号をチェックする事も可能です。

1セクタに対して，リード／ライトの必要なバイト数はIDフィールドを読んだ時のセク

タ長指定バイトによってFDC内で決定されます。セクタ長指定バイトの内容とセクタ長の関係を**表6**に示します。

タイプIIコマンドは，マルチレコードフラグ（m)，ディレイフラグ（E)，サイドフラグ（S)，比較フラグ（C）を持っています。また，ライトデータコマンドでは，アドレスマークフラグ（a_0）を持つ。

マルチレコードフラグ（m)：

m＝1：連続セクタ（セクタ番号が増加する方向）でリード／ライトを行ないます。

m＝0：単一セクタでリード／ライトを行ないます。

ディレイフラグ（E)：

E＝1：HLDを“H”としたのち15ms待ち，HLTをサンプリングします。

E＝0：HLDを“H”としたのち，ただちにHLTをサンプリングします。

比較フラグ（C)：

C＝1：サイド番号の比較を行ないます。

C＝0：サイド番号の比較を行ないません。

サイドフラグ（S)：

S＝1：サイド番号のLSBが1のとき一致したものとみなします。

S＝0：サイド番号のLSBが0のとき一致したものとみなします。

（このフラグは，Cフラグが1のときのみ有効）

アドレススマークフラグ（a_0)：

a_0＝0：データアドレスマークに（FB)$_H$（Data Mark）を書きます。

a_0＝1：データアドレスマークに（F 8)$_H$（Deleted Data Mark）を書きます。

〔表5〕タイプIIコマンド

コマンド名称	コマンドレジスタビット (MSB) 7 6 5 4 3 2 1 0 (LSB)
リードデータ	1 0 0 m S E C 0
ライトデータ	1 0 1 m S E C a_0

〔表6〕セクタ長

セクタ長指定バイト	セクタのデータ量
(00)$_H$	128バイト
(01)$_H$	256 〃
(02)$_H$	512 〃
(03)$_H$	1024 〃

タイプIコマンド処理フローチャート

タイプIコマンド

BUSY←1, CRC ERROR←0
SEEK ERROR←0, IRQ←0
DRQ←0

h＝1?
YES
HLD←1
NO
HLD←0

ステップイン?
YES
DIRC←1
NO

ステップアウト?
YES
DIRC←0
NO

ステップ?
YES
NO

シーク?
YES
(リストア)
NO

u＝1?
NO
YES

TR←(FF)H
DR←(00)H
DSR←(DR)

(DSR)＝(TR)?
YES
NO

(DSR)>(TR)?
YES
DIRC←1
NO
DIRC←0

DIRC＝1?
YES
TR←(TR)＋1
NO
TR←(TR)－1

TROO・DIRC＝1
YES
TR←(00)H
NO
1STEPパルス出力
r_1, r_0による遅延

シークまたはリストア?
YES
NO

V＝1?
NO
YES
HLD←1
15msの待ち

HLT＝1?
NO
YES

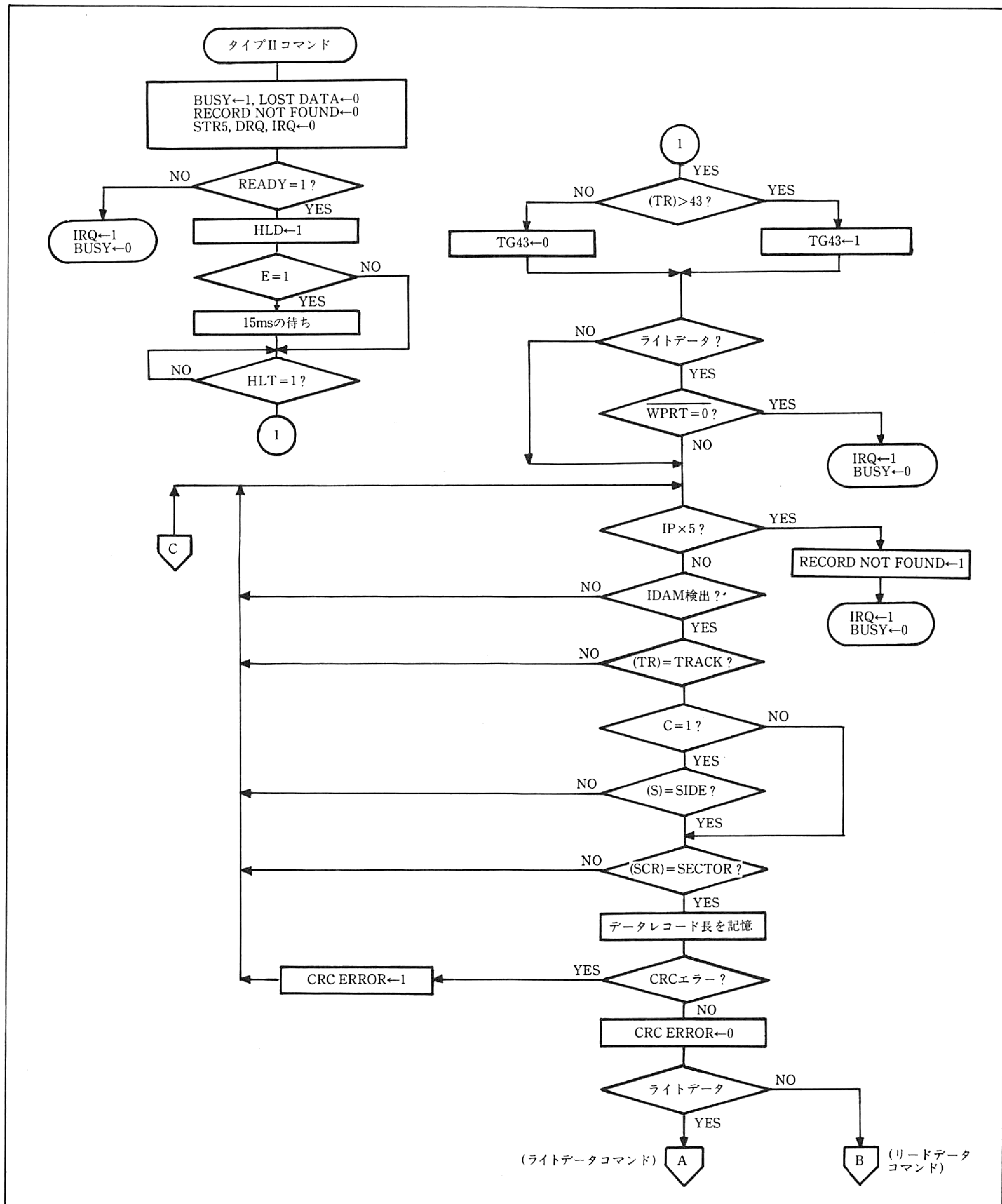
IP×5?
YES
SEEK ERROR←1
NO

IDAM検出?
NO
YES

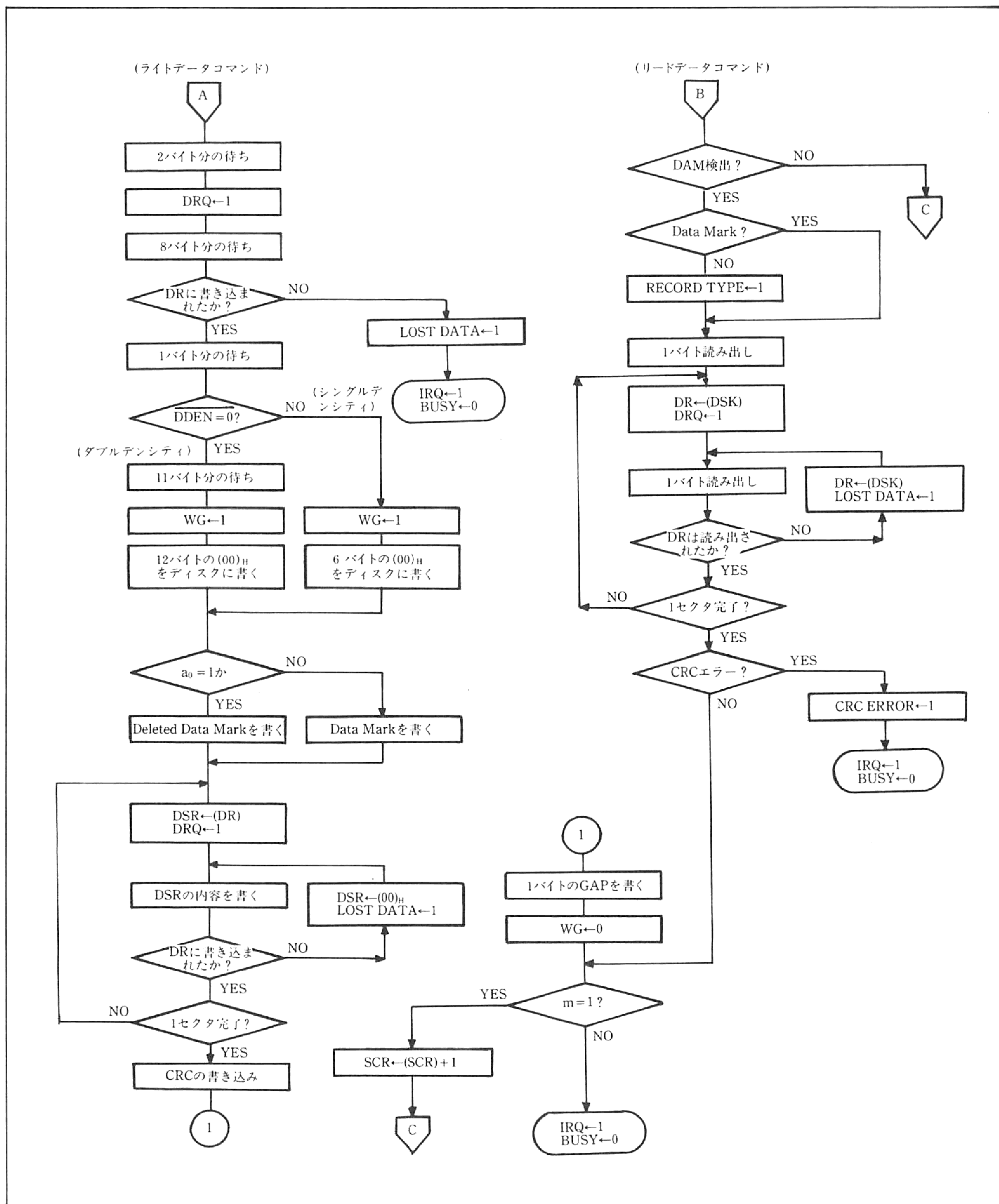
(TR)＝TRACK?
NO
YES

CRCエラー?
YES
CRC ERROR←1
NO
CRC ERROR←0

IRQ←1
BUSY←0


タイプIIコマンド
BUSY←1, LOST DATA←0
RECORD NOT FOUND←0
STR5, DRQ, IRQ←0
NO
READY=1 ?
YES
IRQ←1
BUSY←0
HLD←1
E=1
NO
YES
15msの待ち
NO
HLT=1 ?
1
1
YES
NO
(TR)>43 ?
YES
TG43←0
TG43←1
NO
ライトデータ ?
YES
WPRT=0 ?
YES
NO
IRQ←1
BUSY←0
C
IP×5 ?
YES
NO
RECORD NOT FOUND←1
NO
IDAM検出 ?
YES
IRQ←1
BUSY←0
NO
(TR)=TRACK ?
C=1 ?
NO
YES
NO
(S)=SIDE ?
YES
NO
(SCR)=SECTOR ?
YES
データレコード長を記憶
YES
CRC ERROR←1
CRCエラー ?
NO
CRC ERROR←0
ライトデータ
NO
YES
(ライトデータコマンド)
A
B
(リードデータコマンド)

タイプIIコマンド処理フローチャート(その2)



タイプⅢコマンド

タイプⅢコマンドは，リードアドレス，リードトラック，ライトトラックコマンドがあります。これらのコマンドは，現在ヘッドのあるトラックに対して処理が行なわれます。

〔表7〕タイプⅢコマンド

コマンド名称	コマンドレジスタビット (MSB)　7 6 5 4 3 2 1 0　(LSB)
リードアドレス	1 1 0 0 0 E 0 0
リードトラック	1 1 1 0 0 E 0 0
ライトトラック	1 1 1 1 0 E 0 0

リードアドレスコマンドは最初に出合ったIDフィールドの内容を読み出します。IDフィールドは6バイトであり，その順序と内容は，

1．トラックアドレス（トラック番号）
2．サイドナンバー　（ディスク面番号）
3．セクタアドレス　（セクタ番号）
4．セクタ長　　　　（セクタ長指定バイト）
5．CRC1
6．CRC2　（本IDフィールドのCRC 2バイト）

です。

リードトラックコマンドは，$\overline{\text{IP}}$入力を検出した後ディスク内全データを読み出します。データの同期は，インデックスマーク，IDアドレスマーク，データアドレスマークで行なわれます。

ライトトラックコマンドでは，最初の1バイトがデータレジスタに書き込まれてから，$\overline{\text{IP}}$を検出し，第1バイトを書き込みます。これに続いてディスクに書き込むべきデータを1バイトづつデータレジスタに転送しなければなりません。もし，1バイトのデータがディスクに書き込まれてもデータレジスタに書き込み用データの転送を怠ると，$(00)_H$が書き込まれたものとし，ディスクへ書き込む。この動作を次の$\overline{\text{IP}}$が検出されるまで繰り返します。

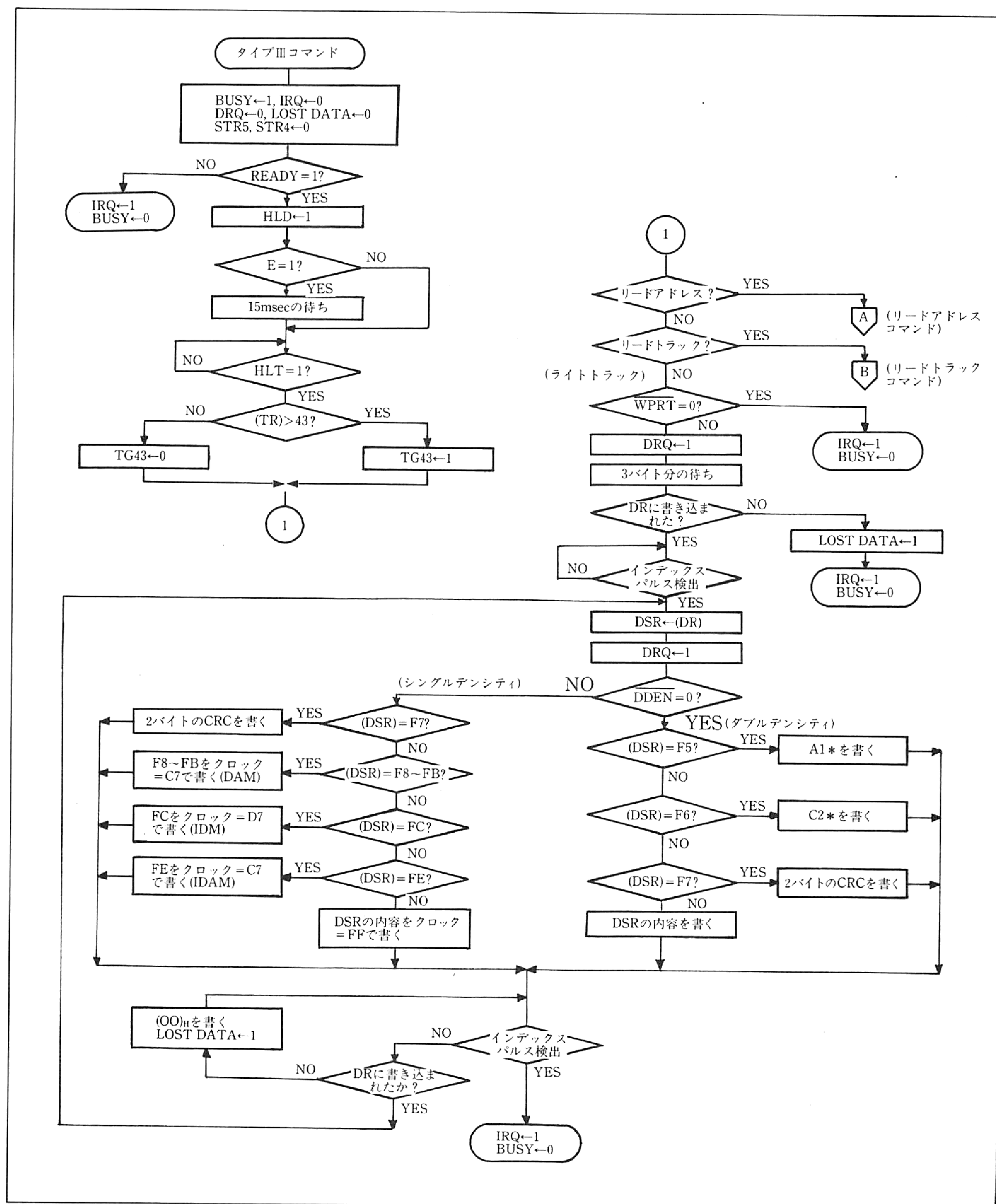
また，このコマンドではディスクイニシャライズのために，$(F5)_H$～$(FE)_H$がデータレジスタに書き込まれると，特別の処理を行ないます。これらのデータに対するFDCの動作を**表8**に示します。

〔表8〕ディスクイニシャライズのためのデータ

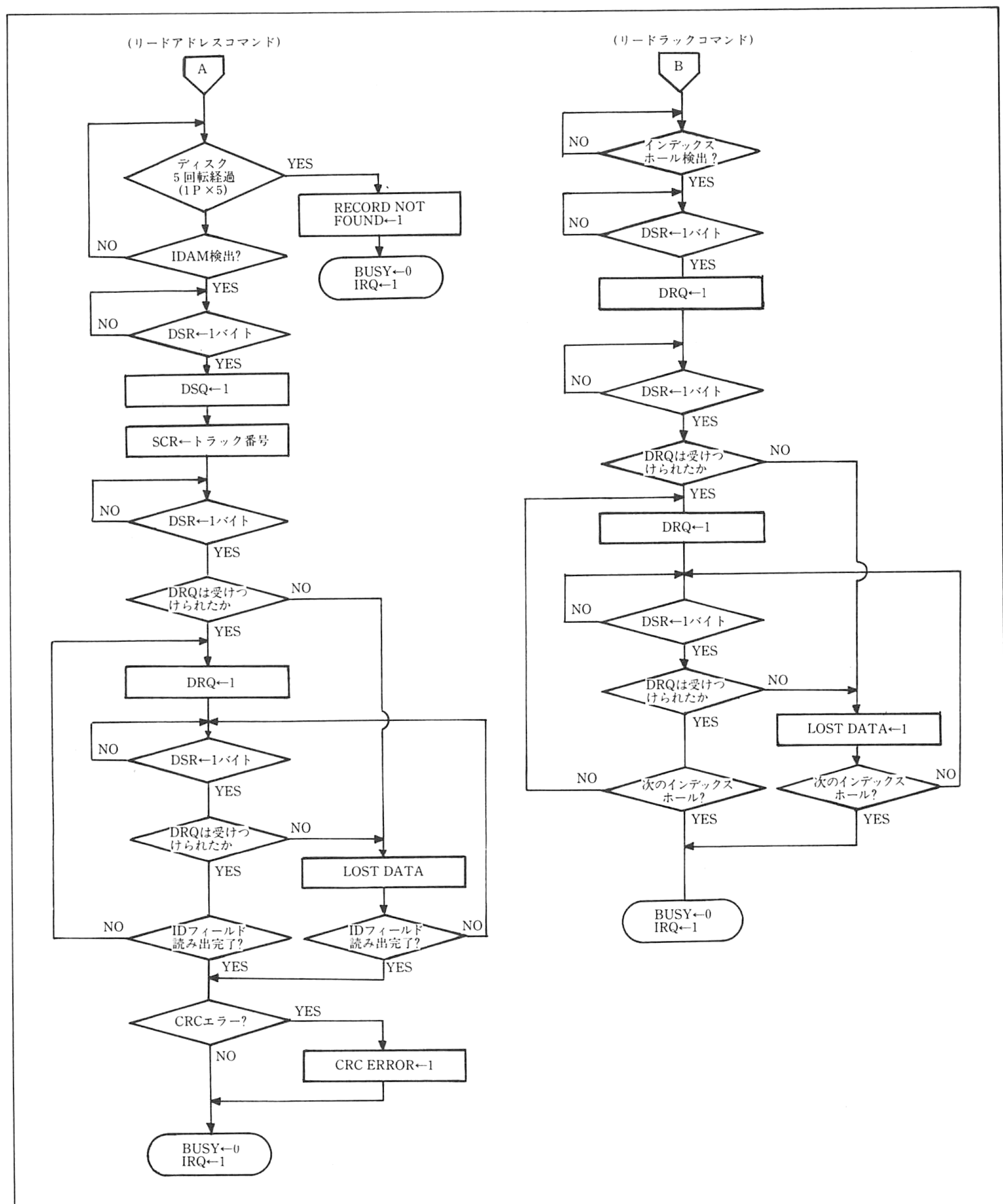
DRの内容	FMの場合（$\overline{\text{DDEN}}$＝“H”）			MFMの場合（$\overline{\text{DDEN}}$＝“L”）	
	ディスクに書き込むデータ		DRの意味	ディスクに書き込むデータ	DRの意味
	データパターン	クロックパターン			
00 ～ F4	00 ～ F4	FF ～ FF	データ ～ データ	00 ～ F4	データ ～ データ
F5			(禁止)	A1*	IDAM, DAMの前提データ
F6			(禁止)	C2*	IDMの前提データ
F7	CRC1, CRC2	FF	内部で計算されたCRC 2バイトを書く。	CRC1, CRC2	内部で計算されたCRC 2バイトを書く。
F8	F8	C7	DAMとしてDDMを書く。内部ではCRCをプリセットする。	F8	データ（注1）
F9 FA	F9 FA	C7 C7	内部ではCRCをプリセットする。	F9 FA	データ データ
FB	FB	C7	DAMとしてDMを書く。内部ではCRCをプリセットする。	FB	データ（注2）
FC	FC	D7	IDMを書く。	FC	データ（注3）
FD	FD	FF	データ	FD	データ
FE	FE	C7	IDAMを書く。内部ではCRCをプリセットする。	FE	データ（注4）
FF	FF	FF	データ	FF	データ

(注1) A1* 3バイトに続けて書かれた場合，デリーテッドデータマークとなる。
(注2) A1* 3バイトに続けて書かれた場合，データマークとなる。
(注3) C2* 3バイトに続けて書かれた場合，インデックスマークとなる。
(注4) A1* 3バイトに続けて書かれた場合，IDアドレスマークとなる。

タイプIIIコマンド処理フローチャート(そのI)



タイプIIIコマンド処理フローチャート(その2)



タイプIVコマンド

タイプIVコマンドは，フォースインタラプトコマンドです。このコマンドはいつもコマンドレジスタに書き込む事により実行されます。(他のコマンドは，実行中のコマンドがある時に新たにコマンドを発すると，その動作は保障されません)

フォースインタラプトが書き込まれると，指定に応じた条件でIRQが“H”レベルになります。もし，このコマンドが書き込まれた時，他の実行中のコマンドがあると，実行中のコマンドは打ち切られ，フォースインタラプトコマンドの実行を行ないます。

表9に，フォースインタラプトコマンドのコード，**表10**にIRQ発生条件を示す。

〔表9〕タイプIVコマンド

コマンド名称	**コマンドレジスタビット** (MSB)　7 6 5 4 3 2 1 0　(LSB)
フォース インタラプト	1 1 0 1 I_3 I_2 I_1 I_0

〔表10〕IRQ発生条件

I_i	IRQ発生条件
$I_0 = 1$	REDY入力の立ち上りでIRQ発生(IRQ＝“H”
$I_1 = 1$	REDY入力の立ち下りでIRQ発生
$I_2 = 1$	各インデックスパルス($\overline{\mathrm{IP}}$)でIRQ発生
$I_3 = 1$	無条件でただちにIRQ発生

ステイタス

FDCは，コマンドの実行結果，実行中の状態を示すためにステイタスレジスタがあります。コマンドが実行されると，所要のビットがプリセットされ，コマンド実行中に更新され，コマンド実行終了直前に内容が確定します。各ビットの示す意味は，実行中のコマンド，実行終了したコマンドにより変化します。ステイタスレジスタの内容を**表11**に示し，各ステイタスの説明を**表12**に示します。

〔表11〕ステイタスレジスタの表示

コマンド ＼ ステイタスレジスタ		STR7	STR6	STR5	STR4	STR3	STR2	STR1	STR0
TYPE I	すべてのコマンド	NOT READY	WRITE PROTECT	HEAD ENGAGED	SEEK ERROR	CRC ERROR	TRACK00	INDEX	BUSY
TYPE II	リードデータ	NOT READY	0	RECORD TYPE	RECORD NOT FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
	ライトデータ	NOT READY	WRITE PROTECT	WRITE FAULT	RECORD NOT FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
TYPE III	リードアドレス	NOT READY	0	0	RECORD NOT FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
	リードトラック	NOT READY	0	0	0	0	LOST DATA	DATA REQUEST	BUSY
	ライトトラック	NOT READY	WRITE PROTECT	WRITE FAULT	0	0	LOST DATA	DATA REQUEST	BUSY
TYPE IV	(他のコマンド実行中の場合)	(今まで実行していたコマンドのステイタスビットと同様の意味)							0
	(実行中のコマンドのない場合)	NOT READY	WRITE PROTECT	HEAD ENGAGED	0	0	TRACK00	INDEX	0
マスタリセット		(TYPE I コマンドに準ずる)							

〔表12〕ステイタスの意味

コマンド	ステイタス	意　　味
タイプIコマンド	NOT－READY (STR7)	NOT－READY＝1でディスクドライブが動作可能状態でない事を示します。これは，$\overline{\text{READY}}$とMRの論理です。
	WRITE－PROTECT (STR6)	WRITE－PROTECT＝1でディスクへの書き込みが禁止されている事を示します。これは，$\overline{\text{WPRT}}$入力の反転コピーです。
	HEAD－ENGAGED (STR5)	HEAD－ENGAGED＝1でヘッドがメディアに押しつけられている事を示します。これはHLDとHLTの論理積です。
	SEEK－ERROR (STR4)	SEEK－ERROR＝1でベリファイ動作が成功しなかった事を示します。これは，IDフィールドが検出されなかったか，IDフィールドのトラック番号とトラックレジスタの内容が一致しない事によります。
	CRC－ERROR (STR3)	CRC－ERROR＝1で，IDフィールド読み出し時に読み出しエラーがあった事を示します。
	TRACK00 (STR2)	TRACK00＝1でディスクのヘッドがトラック0の上にある事を示します。TRACK00は$\overline{\text{TR00}}$入力の反転コピーです。
	INDEX (STR1)	INDEX＝1でインデックスホールを検出した事を示します。これはIP入力の反転コピーです。
	BUSY (STR0)	BUSY＝1でFDCがコマンド動作中である事を示します。
タイプII／IIIコマンド	NOT－READY (STR7)	NOT－READY＝1でディスクドライブが動作可能状態でない事を示します。これは$\overline{\text{READY}}$とMRの論理和です。
	WRITE－PROTECT (STR6)	WRITE－PROTECT＝1でディスクへの書き込みが禁止されている事を示します。($\overline{\text{WPRT}}$の反転コピー)
	RECORD TYPE /WRITE FAULT (STR5)	リード動作の時，RECORD TYPEの表示として，DAMがDDMの時セットされます。 ライト動作の時，WRITE FAULTの表示として，書き込み動作が打ち切られた事を示します。 この時ステイタスは$\overline{\text{WF}}$の反転コピーでラッチされます。
	RECORD NOT－FOUND (STR4)	RECORD－NOT－FOUND＝1で指定されたトラック番号，サイド番号，セクタ番号を持ち，正しく読み出されたIDフィールドがなかった事を示します。
	CRC－ERROR (STR3)	CRC－ERROR＝1でディスクからIDフィールドもしくはデータフィールドの読み出しにおいて，読み出しエラーがあった事を示します。
	LOST－DATA (STR2)	LOST－DATA＝1で所要時間内にDRの読み出し，もしくは書き込みが行なわれなかったデータがあった事を示します。
	DATA－REQUEST (STR1)	DATA－REQUEST＝1で，FDCがプロセッサに対してDRの読み出し，もしくは書き込みを要求している事を示します。これは，DRQのコピーです。
	BUSY (STR0)	BUSY＝1でFDCがコマンド実行中である事を示します。

IRQ，DRQ リセット

FDC はコマンド終了，打ち切りを IRQ で通知しますが，この出力はステイタスレジスタの読み出し，もしくは，新コマンドの書き込みでリセットされます。 ただし，タイプⅣコマンドによってセットされた IRQ は，I_0～I_3＝0 のタイプⅣコマンドの書き込み後，ステイタスレジスタの読み出し，もしくは新コマンドの書き込みによってリセットされます。

DRQ はデータレジスタの読み出し，もしくは，書き込みによってリセットされます。

ディスクフォーマッティング

ディスクのフォーマッティングは，FDC にライトトラックコマンドを書き込み，DRQ が"H"レベルになる毎に，データレジスタにデータを書き込んで行ないます。シングルデンシティ，ダブルデンシティの各フォーマット時のデータ転送例を示します。

(1)IBM 3740フォーマット

	バイト数	データ	
	40	(FF)$_H$	
	6	(00)$_H$	
	1	(FC)$_H$	－インデックスマーク
	26	(FF)$_H$	
	6	(00)$_H$	
	1	(FE)$_H$	
	1	トラック番号	(00)$_H$～(4C)$_H$
	1	サイド番号	(00)$_H$ or (01)$_H$
	1	セクタ番号	(01)$_H$～(1A)$_H$
*	1	セクタ長	(00)$_H$
	1	(F7)$_H$	－CRC 2バイト
	11	(FF)$_H$	
	6	(00)$_H$	
	1	(FB)$_H$	－データマーク
	128	データ	(E5)$_H$
	1	(F7)$_H$	－CRC 2バイト
	27	(FF)$_H$	
	247**	(FF)$_H$	

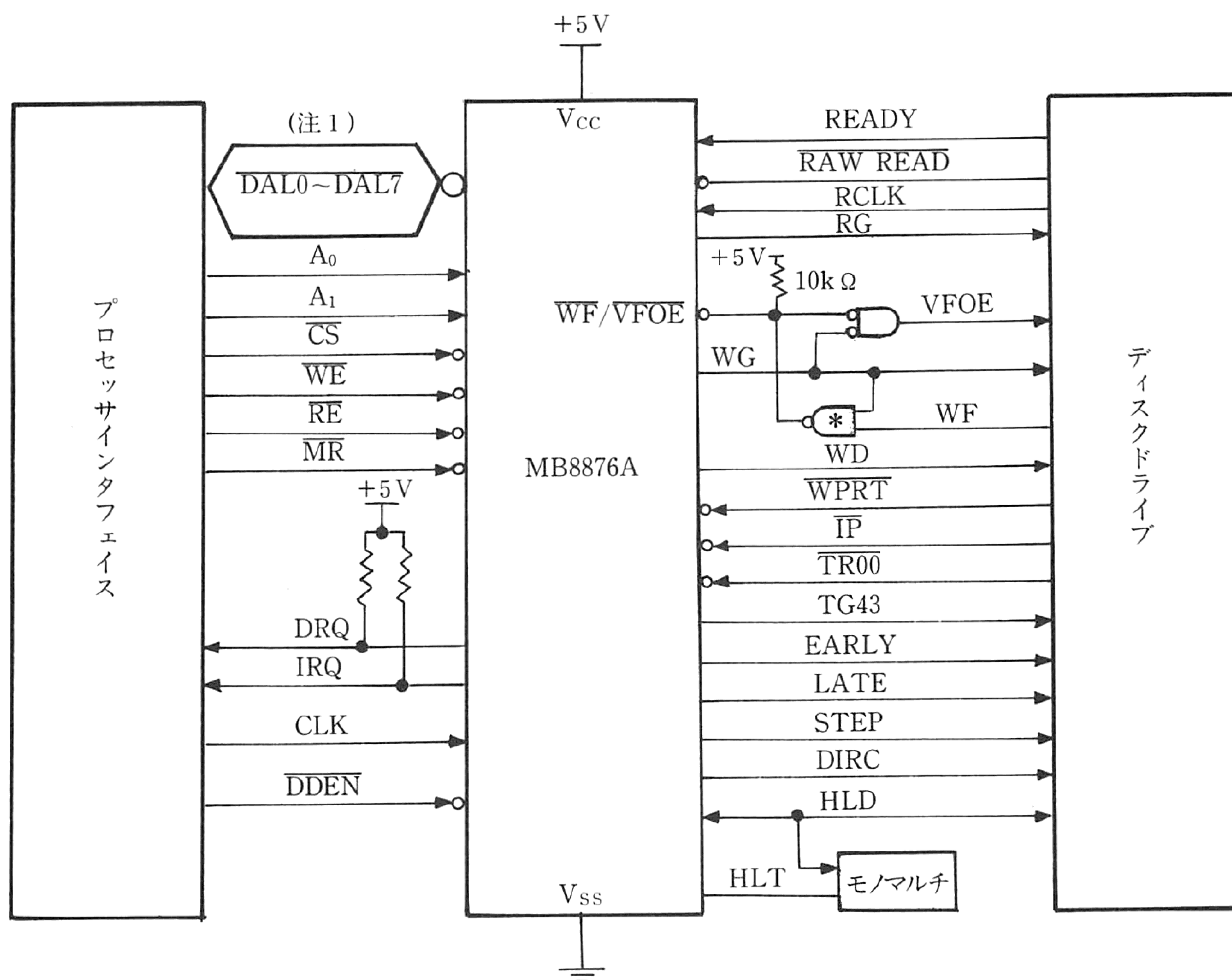
（注）　*　このサイクルをセクタ番号を＋1しながら26回繰り返します。

　　　**　このバイト数は標準値です。これをIRQ が"H"となるまで転送します。

(2) IBMシステム34フォーマット

バイト数	データ
80	$(4E)_H$
12	$(00)_H$
3	$(F6)_H$
1	$(FC)_H$ －インデックスマーク
50	$(4E)_H$
12	$(00)_H$
3	$(F5)_H$
1	$(FE)_H$
1	トラック番号 $(00)_H-(4C)_H$
1	サイド番号 $(00)_H$ or $(01)_H$
1	セクタ番号 $(01)_H-(1A)_H$
1	セクタ長 $(01)_H$
* 1	$(F7)_H$ －CRC 2バイト
22	$(4E)_H$
12	$(00)_H$
3	$(F5)_H$
1	$(FB)_H$ －データマーク
256	データ $(40)_H$
1	$(F7)_H$ －CRC 2バイト
54	$(4E)_H$
598**	$(4E)_H$

(注)　*　このサイクルをセクタ番号を＋1しながら26回繰り返します。
　　**　このバイト数は標準値です。これをIRQが，“H”になるまで転送します。



注1．MB8877Aの場合，DAL0~DAL7は正論理となる。
　＊　オープンコレクタ出力

付 録 2

μPD765A/745/7265仕様書

本書中で765と称しているのはμPD765，μPD765A の両方どちらにも適応される。

μPD765及びμPD7265はμPD765のECMAフォーマットのバージョンである。

μPD765A/745/7265　EDCユーザーズマニュアル
（1983年 2 月24日版）日本電気刊　より抜粋

1 概　説

1－1 特　　徴

■ μPD765A

・IBMディスケット1　（片面128，256カレイ，512バイト／セクタ）コンパチブル

・　　　〃　　　2　（片面256バイト／セクタ）コンパチブル

・　　　〃　　　2D（両面倍密度256，1024バイト／セク　タ）コンパチブル

■ μPD745／7265

・ECMA／ISO ミニフロッピィ・フォーマット・コンパチブル

・ECMA66（ISO／TC 97／SC11 N419）…片面単密度（256バイト／セクタ）

・ECMA70（ISO／TC 97／SC11 N475）…両面倍密度（256バイト／セクタ）

■ μPD765A，μPD745／7265

・FM，MFM 制御（コマンドで指定）

・記録長は可変；128，256，512，1024，2048，4096，8192バイト／セクタ

・マルチセクタ機能

・マルチトラック機能

・フロッピィ・ディスク・ドライブを4台まで接続可能

・シーク動作は同時に4台まで可能

・セクタの途中までリード／ライト指定可能（128バイト／セクタ＝FM 時のみ）

・CRC 発生，チェック機能内蔵（$X^{16}+X^{12}+X^{5}+1$）

・ステップ速度プログラマブル

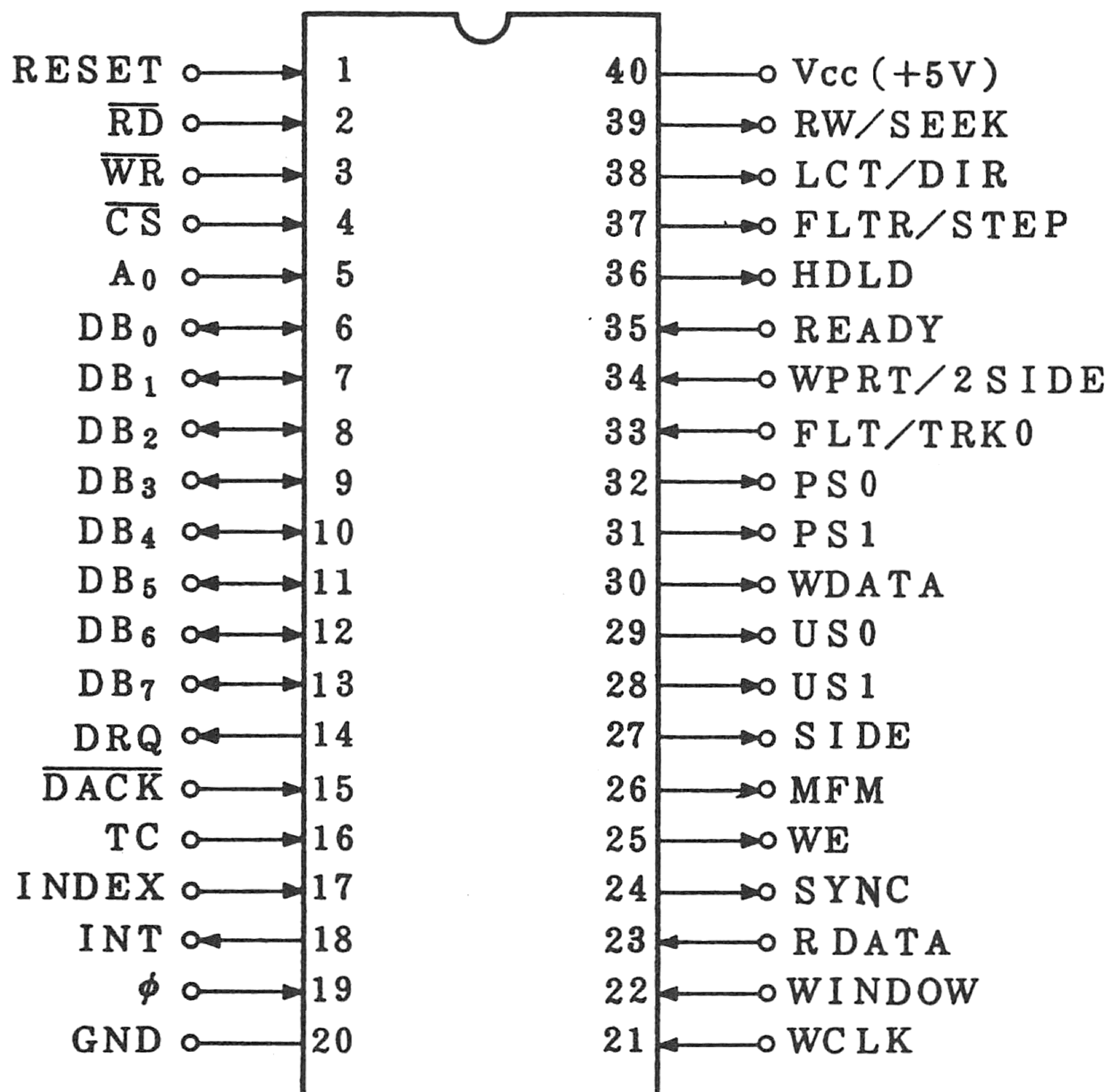
・ヘッドロード時間，ヘッドアンロード時間プログラマブル

・データスキャン機能（一致，大，小の検出）

・DMA/Non-DMA（Interrupt）データ転送

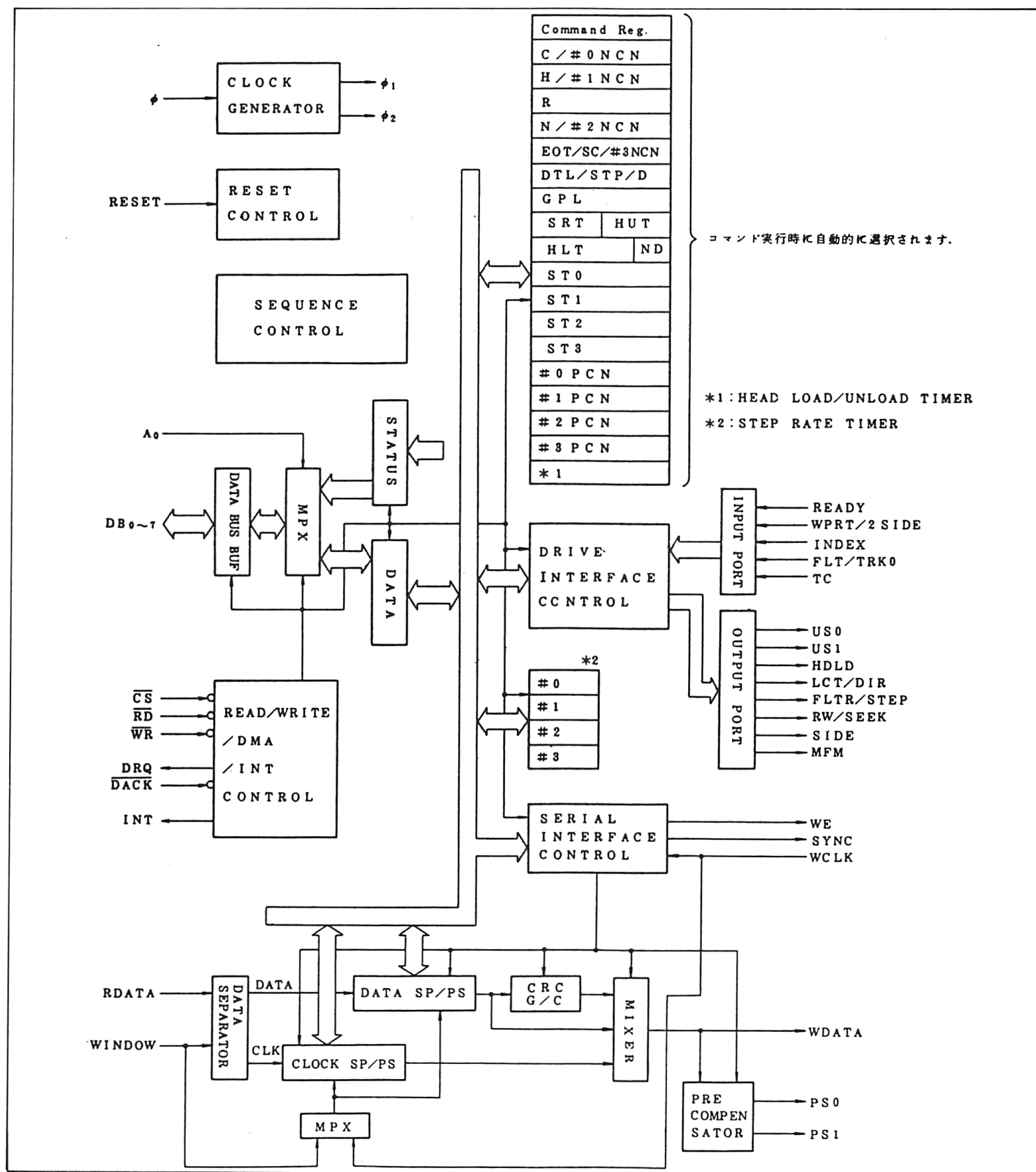
・ライトデータ補正用制御信号発生
・8080系データ，制御バスコンパチブル
・１相クロック　8MHz（標準フロッピィ・ディスク），4MHz（ミニフロッピィ）
・Nチャネル MOS
・単電源　＋５Ｖ
・40ピン DIP

1-2　μPD765A/745/7265端子接続図



RESET : Reset
$\overline{RD}$: Read
$\overline{WR}$: Write
$\overline{CS}$: Chip Select
A_0 : A_0
$DB_{0\sim7}$: Data Bus
DRQ : DMA Request
$\overline{DACK}$: DMA Acknowledge
TC : Terminal Count
INDEX : Index
INT : Interrupt Request
ϕ : Clock
GND : Ground
WCLK : Write Clock
WINDOW : Data Window
RDATA : Read Data
SYNC : VFO Synchronize
WE : Write Enable
MFM : MFM Mode
SIDE : Side Select
US0, 1 : Unit Select
WDATA : Writ Select Date
PS0, 1 : Pre-Shift
FLT : Fault
TRK0 : Track 0
WPRT : Write Protected
2SIDE : Two Side
READY : Ready
HDLD : Head Load
FLTR : Fault Reset
STEP : Step
LCT : Low Current
DIR : Direction
RW/SEEK : Read Write/Seek

1-3 μPD765A/745/7265ブロック図



2 μPD765A/745/7265の端子機能

端子名	I/O	機能
V_{CC}	—	＋５Ｖ電源
GND	—	Ground
ϕ	I	単相，ＴＴＬレベルクロックで，330Ωのプルアップ抵抗を必要とします。標準フロッピィ：8MHz，ミニフロッピィ：4MHz
RESET	I	ＦＤＣをアイドル状態にし，PS0,1とWDATA出力(不定)を除くドライブ・インタフェース出力をロウにします。さらにINT，DRQ出力もロウにします。ＤＢは入力状態となります。
$\overline{CS}$	I	$\overline{RD}$，$\overline{WR}$信号を有効にします。
DB_7〜DB_0	I/O	双方向，３ステートのデータバスです。
$\overline{WR}$	I	データバスを介してＦＤＣへデータを書込むための制御信号です。
$\overline{RD}$	I	データバスを介してＦＤＣからデータを読出すための制御信号です。
INT	O	ＦＤＣがサービスを要求している事を示す信号です。 NON DMAモード時には１バイトごとに，ＤＭＡモード時にはコマンド動作の終了時に出力されます。
A_0	I	データバスを介してアクセスするＦＤＣ内のステータスレジスタとデータレジスタを選択するための信号です。０のときステータスレジスタ，１のときデータレジスタを選択します。
DRQ	O	ＤＭＡモードによるＦＤＣとメモリとのデータ転送要求信号です。
$\overline{DACK}$	I	ＤＭＡサイクルが与えられていることを示す信号です。
US0,1	O	ドライブ選択信号です。これをデコードすることにより，４台までのＦＤＤの選択信号となります。
MFM	O	ＶＦＯ回路の動作モードを指定する信号です。１のときＭＦＭモード，０のときＦＭモードを指定します。

端　子　名	I/O	機　　　　能
SYNC	O	ＶＦＯ回路の動作モードを指定する信号です。１のときリード動作を許可し，０のときリード動作を禁止します。
RW/SEEK	O	ドライブ・インタフェース信号の内，リード／ライト用とシーク用を兼用している信号を区分けるための信号です。０のときＲＷ，１のときSEEKを示します。
HDLD	O	ドライブのリード/ライト・ヘッドをロード状態にする信号です。
SIDE	O	両面型ドライブのヘッド０，ヘッド１を選択する信号です。０のときヘッド０，１のときヘッド１を選択します。
LCT/DIR	O	RW/SEEK信号がＲＷを指定しているときにはＬＣＴとなり，ドライブのリード／ライト・ヘッドが43以上のシリンダを選択していることを示す信号となります。またSEEKを指定しているときにはDIRとなり，シークの向きを指定する信号となります。０のとき遠心方向，１のとき求心方向を指定します。
FLTR/STEP	O	RW/SEEK信号がＲＷを指定しているときにはFLTRとなり，ドライブ側で保持しているFAULT状態をリセットする信号となります。またSEEKを指定しているときにはSTEPとなり，シークステップ信号となります。
READY	I	ドライブがレディ状態であることを示す信号です。
WPRT/2SIDE	I	RW/SEEK信号がＲＷを指定しているときにはWPRTとなり，ドライブまたはメディアが書込み禁止状態であることを示す信号となります。またSEEKを指定しているときには2SIDEとなり，両面用のメディアが入っていることを示す信号となります。
INDEX	I	メディア上のトラックの物理的開始点を示す信号です。
FLT/TRK0	I	BW/SEEK信号がＲＷを指定しているときにはＦＬＴとなり，ドライブがFAULT状態であることを示す信号となります。またSEEKを指定しているときにはTRK０となり，リード/ライト・ヘッドがシリンダ０に位置していることを示す信号となります。
TC	I	メインシステムからのリードまたはライト動作の終了指示信号です。
WDATA	O	ドライブへの書込みデータで，クロックビットとデータビットから構成されている信号です。

<table>
<tr><th>端子名</th><th>I/O</th><th>機能</th></tr>
<tr><td>WE</td><td>O</td><td>ドライブに対して書込みを指示する信号です。</td></tr>
<tr><td>WCLK</td><td>I</td><td>ドライブへの書込みデータのタイミング信号です。標準フロッピィに対してFMモードのときは500kHz，MFMモードのときは1MHzとし，ミニフロッピィに対してFMモードのときは250kHz，MFMモードのときは500kHzとします。</td></tr>
<tr><td>PS0, 1</td><td>O</td><td>MFMモードで書込むとき，リード時のマージンを得るため，書込みデータを一定時間早めるか，遅らせるかを指定する信号です。WDATA信号を次表のように制御指定します。
<table>
<tr><th>PS0</th><th>PS1</th><th>FM</th><th>MFM</th></tr>
<tr><td>0</td><td>0</td><td>そのまま</td><td>そのまま</td></tr>
<tr><td>0</td><td>1</td><td>—</td><td>LATE
225～250ns</td></tr>
<tr><td>1</td><td>0</td><td>—</td><td>EARLY
225～250ns</td></tr>
<tr><td>1</td><td>1</td><td>—</td><td>—</td></tr>
</table></td></tr>
<tr><td>RDATA</td><td>I</td><td>ドライブから読出しデータで，クロックビットとデータビットから構成されている信号です。</td></tr>
<tr><td>WINDOW</td><td>I</td><td>VFO回路において作られる信号で，RDATAをサンプルするために使用されます。RDATAのデータビットとWINDOW信号との位相同期はFDC内で行われます。</td></tr>
</table>

3 μPD765A/745/7265のレジスタ構成

3－1　メインシステム側インタフェース用

FDC内にメインシステムとのインタフェース用レジスタとして，データレジスタ（DATA）とステータスレジスタ（STATUS）を持っています。各レジスタなA$_0$，$\overline{RD}$，$\overline{WR}$制御信号によって選択され，それらの関係は**表3－1**の通りです。

〔表3－1〕

A_0	$\overline{RD}$	$\overline{WR}$	動　　　作
0	0	1	ステータス・レジスタ・リード
	×	0	禁　止
1	0	1	データ・レジスタ・リード
	1	0	データ・レジスタ・ライト

注）$\overline{DACK}$入力がアクティブ（0）のときは$\overline{CS}$，A0の状態にかかわらずデータ・レジスタが選択されます。

(1)　データレジスタ（DATA）

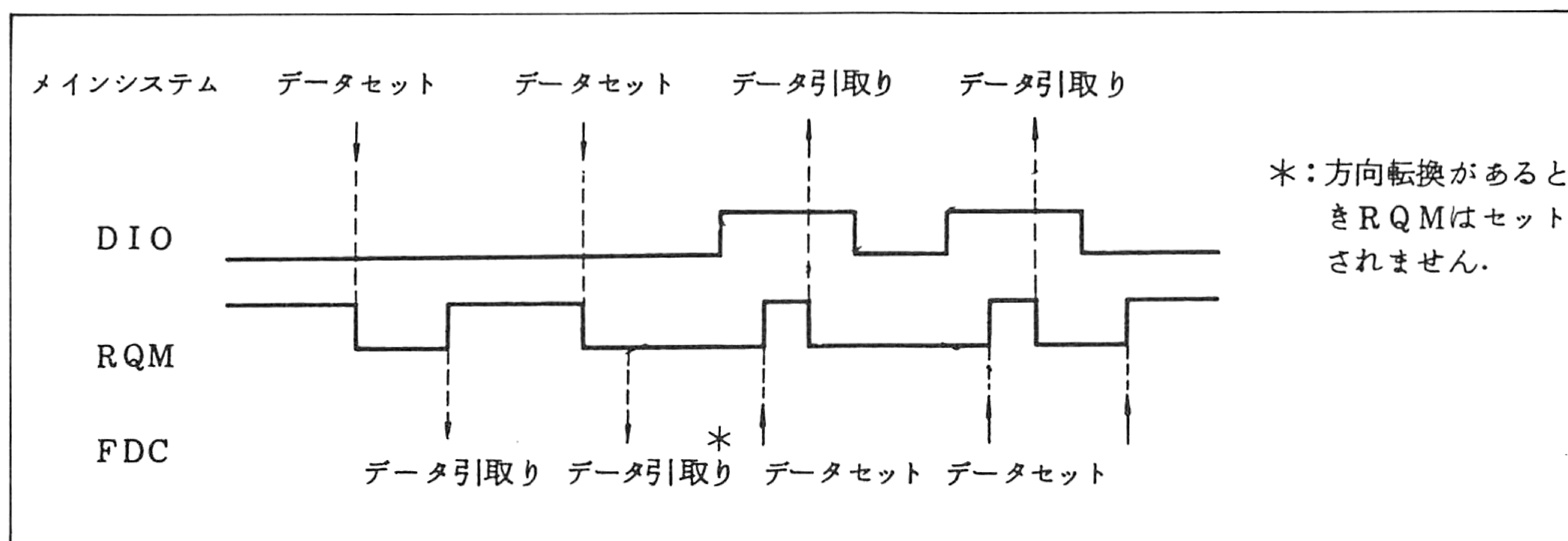
FDCとメインシステム間で転送する各種情報（コマンド，パラメータ，データ，およびリザルトステータス）を一時的にストアする8ビットレジスタです。

(2)　ステータスレジスタ（STATUS）

FDCの状態を示す8ビットレジスタで，その構成を**表3－2**に示します。メインシステムは任意の時点でその内容を読取ることができます。

〔表3-2〕ステータスレジスタ

ビット番号	名称	略称	内容
D0	FD0 Busy	D0B	デバイス＃0がSEEKコマンドによるシーク動作を実行中であるか，シーク動作終了の割込み要求を保留中であることを示します。
D1	FD1 Busy	D1B	デバイス＃1についてＤ０ビットの内容と同様。
D2	FD2 Busy	D2B	デバイス＃2についてＤ０ビットの内容と同様。
D3	FD3 Busy	D3B	デバイス＃3についてＤ０ビットの内容と同様。
D4	FDC Busy	CB	FDCがCommand Phase, Result Phase, またはリード／ライト・コマンドのExecution Phaseを実行中であることを示します。このビットがセットされているときは，他のコマンドは受付けられません。
D5	Non-DMA MODE	NDM	FDCがNon-DMAモードでデータ転送中であり，メインシステムに対してサービスを要求していることを示します。
D6	Data Input/Output	DIO	データレジスタを介して転送するデータの方向を示します。0のときはメインシステムからＦＤＣの方向，1のときはＦＤＣからメインシステムの方向を示します。なお，データレジスタの状態はＲＱＭ(Ｄ７ビット)が示します。
D7	Request for Master	RQM	ＦＤＣからメインシステムへ転送すべきデータがデータレジスタにロードされていること，またはデータレジスタが空で，メインシステムからＦＤＣへ転送するデータをデータレジスタに書込んでもよいことを示します。データの方向を示すＤＩＯ(Ｄ６ビット)の状態により，次の働きをします。 DIO＝0のとき： 　メインシステムからＦＤＣへデータを転送する場合で，メインシステムがＦＤＣのデータレジスタにデータをセットしたとき($\overline{WR}$＝0)にＲＱＭは0となり，ＦＤＣがそのデータを引取ったとき1となります。 DIO＝1のとき： 　ＦＤＣからメインシステムへデータ転送する場合で，ＦＤＣがデータレジスタにデータをセットしたとき1となり，メインシステムがそのデータを引取ったとき($\overline{RD}$＝0)0となります。



3－2　シリアルデータ（FDD）・インタフェース用

(1)　データセパレータ (DATA SEPARATOR)

リードモードにおいてWINDOW信号により，リードデータをデータとクロックとに分離します。

分離されたデータはデータ・シフトレジスタへ，クロックはクロック・シフトレジスタへ送られます。

(2)　データ・シフトレジスタ（DATA　SP／PS）

リードモードのときデータセパレータによって分離されたデータを直列から並列に変換して8ビット内部バスに出力し，ライトモードのとき内部バスから入力される8ビット並列データを直列に変換し，CRCジェネレータ／チェッカと出力ミクサに送ります。

(3)　クロック・シフトレジスタ（CLOCK　SP／PS）

リードモードのときデータセパレータによって分離されたクロックを直列から並列に変換して8ビット内部バスに出力し，ライトモードのとき内部バスから入力される8ビット並列データを直列に変換し，出力ミクサに送ります。

(4)　CRCジェネレータ／チェッカ（CRC　G／C）

CRCバイトの生成，チェックを行ないます。CRC生成多項式は次の通りです。

$$X^{16}+X^{12}+X^{5}+1$$

リードモードのときはリードデータのCRCを計算し，リードデータの最後に付加されているCRCバイトと比較を行い，不一致のときはエラーの指示を行ないます。

ライトモードのときはCRCを計算し，転送データの最後に2バイトのCRCバイトを加えます。

(5) プリコンペンセータ（PRECOMPENSATOR）

メディアの特性により，リードデータは書込み時のタイミングよりずれて再生されることがわかっています。またこのずれはデータパターンによって予想できるものであるため，書込み時に，読出し時に予想されるシフト方向とは逆にライトデータをシフトさせて書込むことにより，読出し時のずれを相殺することができます。このずれを補正することは，データウインドウ幅がFMモードに比べて1／2と狭くなるMFMモードではより重要となります。

プリコンペンセータは，MFMモードでデータの書込みを行う場合，ライトデータをシフトさせるための遅延回路（外付）を制御するプリシフト信号（PS0，1）をライトデータのパターンに応じて発生する回路です。

なお，リードデータのずれによる読出し時のマージン不足を問題にしない位精度の高いDATA WINDOW信号を発生するVFO回路を用意すれば，プリシフト機能は不要となります。

4　μPD765A/745/7265の制御

図4－1はμPD765A/745/7265(FDC)を制御するための概略フローチャートを示しています。まず，Command Phase（以後C－Phaseと略す）のスタートにあたってFDCがBusyかどうかみるためにFDCのステータスレジスタを読出し，CB（FDC Busy）ビットをチェックします。さらに各デバイスがシーク動作中でないことを確認します。BusyであればNon-Busyになるまで待つか，別のプロセスに行きます。Busyでなければ，C－Phaseに入り，まずコマンドコードを書込みます。次にステータスレジスタを読出してRQM（Request for Master）ビットとDIO（Data Input／Output）ビットをチェックします。

RQMがオンでDIOがオフならば，FDCが次のパラメータを受取る準備ができていることを示していますので次のパラメータを書込みます。各コマンドの指定順序に従ってC－Phaseのすべてのパラメータを書込むとC－Phaseは終了しますが，その後のフローはコマンドの種類によって異なっています。

SENSE INTERRUPT STATUSおよびSENSE DEVICE STATUSコマンドおよびInvalidコマンドの場合はExecutuion Phase（以後E－Phaseと略す）がありませんので，C－Phaseの書込み終了後ステータスレジスタを読出してRQMとDIOをチェックします。ここでRQM，DIOともにオンであれば，Result Phase（以後R－Phaseと略す）のリザルトステータス情報とパラメータが用意されていることを示していますので，指定の順序に従って読出しを終了するとR－Phaseが終了し，コマンドの終了となります。

SPECIFYコマンドの場合にはC－Phaseのみでコマンドが終了するため，C－Phase終了後別のプロセスに進みます。このときFDCはNon-Busy状態になっています。

SEEKまたはRECALIBRATEコマンドの場合は，C－Phase終了後E－Phaseに入りますが，E－Phaseはメインシステムとは無関係に進められますので，メインシステムはE－Phaseの終了を示すINT要求を持ちます。INT要求が発生したらSENSE INTERRUPT STATRSコマンドを実行し，そのR－PhaseでSEEKまたはRECALIBRATEコマンドの結果をみます。

以上の特殊の命令を除いたリード／ライト・コマンドでは，C－Phaseが終了するとE－Phaseに進みますが，E－PhaseはNon-DMAモードまたはDMAモードのいずれかで処理されます。Non-DMAモードのときは，1バイトデータ転送ごとにINT要求を発生し，メイ

ンシステム側はそれを受けて割込み処理としてデータのリード／ライトを行ないます。

DMAモードのときは，1バイトデータ転送ごとにDRQ要求を発生し，メインシステム側はDMA処理（CPUは介さない）としてデータのリード／ライトを行います。

いずれの場合も指定された分のデータ転送を終了すると，リザルトステータスおよびパラメータの読取りを要求するINT要求が発生されますので，INT要求をみてR－Phaseに入ります。

INT要求が発生するのは次の4つの場合です。

(1) 1バイトごとのデータ・リード／ライト要求（Non-DMA）

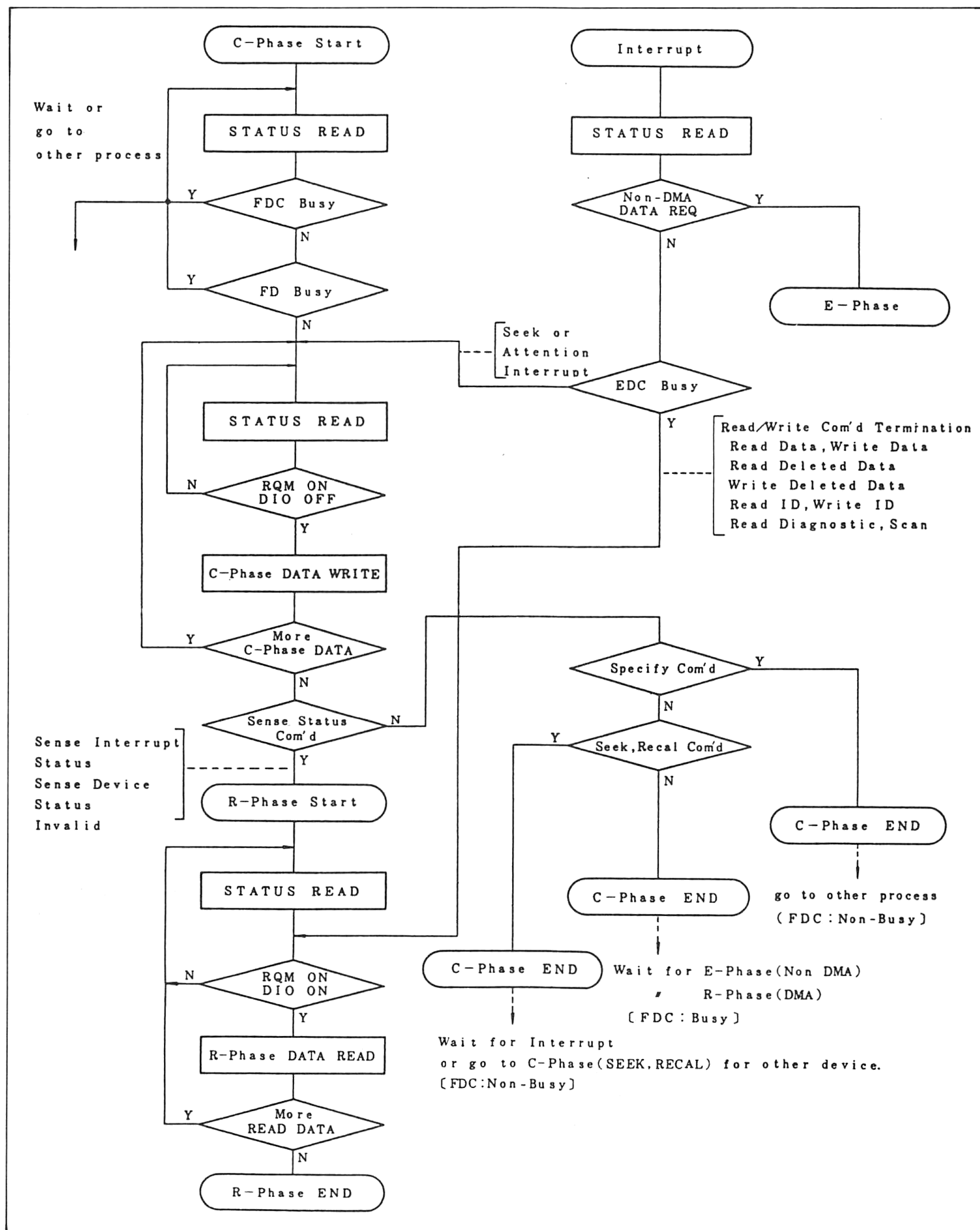
(2) リード／ライトのE－Phase終了によるリザルトステータスの読取り要求

(3) SEEK，RECALIBRATEコマンドのE－Phase終了によるSENSE INTERRUPT STATUSコマンドの起動要求

(4) デバイスの状態遷移によるSENSE INTERRUPT STATUSコマンドの起動要求

したがってINT要求が起きたらいずれによるものかをステータスレジスタを読出してチェックします。ステータスレジスタのNDM（Non-DMA Mode）ビットがオンしていたら，(1)のMon-DMAによるリード／ライト要求であることを示します。NDMビットがオンしていなければ，次にCBビットをチェックしオンならば，(2)のリード／ライト・コマンドのE－PhaseでST0の内容をチェックすることによりわかります。

CBビットがオフ（FDC Non-Busy）ならば，(3)または(4)のSENSE INTERRUPT STATUSコマンドの起動を要求していることを示します。この場合もいずれによるかは，R－PhaseでST0の内容をチェックすることによりわかります。



〔図4-1〕 FDC制御概略フローチャート

5 μPD765A/745/7265のコマンド

5－1 コマンド動作の概要

FDCはメインシステムから与えられるコマンドを次の3つのPhaseに分けて実行します。

（1） Command Phase

FDCがアイドル状態であるときにメインシステムからコマンドが与えられると，続いてそれぞれのコマンドごとに動作を規定するパラメータ情報を受取ります。メインシステム側は，指定されたパラメータを指定された順序に従って転送する必要があります。

（2） Execution Phase

パラメータ情報に従ってコマンドを実行します。ディスク装置とメインシステムとのデータ転送を行います。

DMAまたはINT要求によって処理されます。

（3） Result Phase

コマンドの実行結果を報告するためのリザルトステータス情報などをセットし，メインシステム側がそれらの情報をデータレジスタを介して読出します。必ずINT要求によって処理されます。メインシステム側はすべてのリザルトステータス，パラメータ情報を指定された順序に従って読取る必要があります。

ただし，SEEK，RECALIBRATEコマンドはCommand PhaseとExecution Phaseのみ(INTあり)，SENSE INT STATUS，SENSE DEVICE STATUS，Invalidコマンドは，Command Phase と Result Phaseのみ（INTなし)，SPECIFYコマンドはCommand Phaseのみです。

5－2　コマンドのリザルトステータス情報

Result Phase において読出すべきリザルトステータス情報（ST0～ST3）の内容について**表 5－1～表 5－4** に示します。ST0～ST3 の内容はデータレジスタにセットされ，データバスを介してメインシステム側に読取られます。

たとえば一つの Result Phase で複数のリザルトステータス情報が必要なコマンドに対しては，まず ST0 をデータレジスタにセットし，それが読取られた後 ST1 をセットし，続いて ST2 をセットします。

Result Phase における各リザルトステータスは，必ず全部を読取らないと次のコマンドをセットできません。

〔表5-1〕リザルトステータス0（ST0）

ビット番号	ステータス名称	略称	内容
D7 D6	Interrupt Code	IC	INT要求が何によるかを示します。 D7D6 0 0　コマンドの正常終了（NT） 0 1　コマンドの異常終了（AT） 1 0　起動されたコマンドがInvalidであったため，コマンドを実行しなかった事を示します。(IC) 1 1　デバイスの状態遷移があった事を示します。(AI)
D5	Seek End	SE	SEEKまたはRECALIBATEコマンドによるシーク動作が正常終了または異常終了したときにセットされます。
D4	Equipment Check	EC	デバイスからFault信号を受取ったとき，またはRECALIBRATEコマンド時TRACK 0信号が一定時間内に検出できなかったときセットされます。
D3	Not Ready	NR	指定したデバイスがREADY状態でないときセットされます。
D2	Head Address	HD	INT要求時のヘッドの状態を示します。 SENSE INTERRUPT STATUSコマンド実行時は常に0となっています。
D1	Unit Select1	US1	INT要求時のデバイス番号を示します。
D0	Unit Select0	US0	

NT：Normal Terminate,　AT：Abnormal Terminate,
IC：Invalid Command,　AI：Attention Interrupt

〔表５－２〕リザルトステータス１(ST１)

ビット番号	ステータス名称	略称	内容
D7	End of Cylinder	EN	指定された最終セクタを越えてアクセスを継続させようとしたときセットされます。
D5	Data Error	DE	ディスク上のIDまたはデータを読取った際，CRC エラーを検出するとセットされます。ID，データの区別は，ST2のDDビットによります。
D4	Over Run	OR	あるセクタのデータを処理しているとき，メインシステムのサービスが規定時間内に行われなかったときにセットされます。
D2	No Data	ND	1.次の４種のコマンド実行時にIDRで指定したセクタがトラック上で検出されなかったときセットされます。(READ DATA，WRITE DATA，WRITE DELETED DATA，SCAN) 2.READ IDコマンド実行時トラック上にエラーのないIDが検出されなかったときセットされます。 3.READ DIAGNOSTICコマンド実行時セクタIDとIDRの内容が一致しなかったときセットされます。
D1	Not Writable	NW	WRITE DATA，WRITE DELETED DATA，WRITE IDコマンドを実行時に，書込み不可状態を検出するとセットされます。
D0	Missing Address Mark	MA	1.ディスクのIDをアクセスするコマンドでIM (Index Mark) を２回検出するまでにAM (Address Mark) が検出されなかったときセットされます。 2.ディスクのデータを読取るときDAMまたはDDAMが検出されなかったときセットされます。このときST2のMDビットもセットされます。

備考　D6とD3は使用されず常に０です。

〔表5－3〕リザルトステータス2（ST 2）

ビット番号	ステータス名称	略称	内容
D6	Control Mark	CM	READ DATAまたはSCANコマンド実行時にDDAMの付いたセクタのデータを処理したときセットされます。
D5	Data Error in Data Field	DD	ディスクのデータを読取るときCRCエラーを検出するとセットされます。
D4	No Cylinder	NC	ST 1のNDビットに付帯したステータスで，IDのCバイトが一致しなかったときセットされます。
D3	Scan Equal Hit	SH	SCANコマンドでEqual条件が満足されたときセットされます。
D2	Scan Not Satisfied	SN	SCANコマンドで最終セクタまでスキャンしても条件に合うデータが検出されなかったときセットされます。
D1	Bad Cylinder	BC	ST 1のNDビットに付帯したステータスで，IDのCバイトが一致せずFF(16)であったときセットされます。
D0	Missing Address Mark in Data Field	MD	ディスクのデータを読取るときDAM，またはDDAMが検出されなかったときセットされます。

備考　D7は使用されず常に0です。

〔表5－4〕リザルトステータス3（ST 3）

ビット番号	ステータス名称	略称	内容
D7	Fault	FT	デバイスからのFault信号の状態
D6	Write Protected	WP	デバイスからのWrite Protected信号の状態
D5	Ready	RY	デバイスからのReady信号の状態
D4	Track 0	T0	デバイスからのTrack 0信号の状態
D3	Two Side	TS	デバイスからのTwo Side信号の状態
D2	Head Address	HD	デバイスへのSide Select信号の状態
D1	Unit Select 1	US1	デバイスへのUnit Select 1信号の状態
D0	Unite Select 0	US0	デバイスへのUnit Select 0信号の状態

5-3 コマンド説明に用いられる略称

次に示す略称が以降のコマンドの説明で用いられています。

略　　　　称	意　　　　味
R/W (Read/Write)	リード(R)または(W)信号がアクティブになることを示します。
A_0	コントロール信号A_0の状態を示します。
D_7-D_0 (Data Bus)	8ビットデータ・バスとの対応を示します。
MT (Multitrack)	マルチトラック操作を指定します。MTが1ならマルチトラック操作が許可され，0なら禁止されます。
MF (MFM Mode)	MFMモードかFMモードかの指定をします。1のときMFMモード，0のときFMモードを指定します。
SK (Skip)	Deleted Data Address Markをスキップすることを指定します。
HD (Head)	物理的ヘッド番号0または1を指定します。
US1，US0 (Unit Select)	ドライブユニット番号を示します。
C (Cylinder Address)	メディアのシリンダ番号0～76を示します。
H (Head Address)	論理的ヘッド番号を示します。
R (Record(Sector) Address)	セクタ番号を示します。
N (Record(Sector) Length)	1セクタ内のデータ長を示します。
EOT (End of Track)	あるトラック上の最終セクタ番号を示します。

略　　　　　称	意　　　　　味
GPL (Gap Length)	VFO SYNCを除いたGap 3 の長さ（バイト数）を示します。
ST0, ST1, ST2 (Status)	リザルトステータス情報をストアするレジスタ
ST3 (Status)	ドライブの状態を示す情報をストアするレジスタ
SC (Sector)	トラック当りのセクタ数を示します。
D (Data)	データフィールドに書込むデータのパターンを示します。
STP (Step)	スキャンコマンドを実行するとき，あるセクタが終了したら次のセクタへ進むか，１つあけた次のセクタへ進むかの指定をします。ＳＴＰが１のときは続けて次のセクタをスキャンし，２のときは１つセクタをあけて次のスキャンを行います。
NCN (New Cylinder Number)	シーク操作の結果として新しく選択されるべきシリンダ番号を示します。
PCN (Present Cylinder Number)	SENSE INTERRUPT STATUSコマンドの終了時点におけるシリンダ番号を示します。
SRT (Step Rate Time)	SPECIFY コマンドによって定義されるステップレイト時間を示します。
HUT (Head Unload Time)	SPECIFY コマンドによって定義されるヘッド・アンロード時間を示します。
HLT (Head Load Time)	SPECIFY コマンドによって定義されるヘッド・ロード時間を示します。
ND (Non-DMA Mode)	Non-DMAモードを示し，SPECIFY コマンドによって指定されます。

5-4 コマンド一覧

各コマンドのコマンドフェイズにおける書込み順序，リザルトフェイズにおける読取り順序は，この一覧に記述されている順序に従ってください。

C ： Command Phase

E ： Execution Phase

R ： Result Phase

コマンド		R/WA$_0$	D$_7$ ――― D$_0$	備考
READ DATA				SK：Skip DDAM
	C	W1	MT MF SK 0 0 1 1 0	MT：マルチトラック，MF：MFMモード
	C		× × × × × HD US1 US0	HD：ヘッド番号，US：デバイス番号
	C		←― C ―→	実行開始セクタのID情報
	C		←― H ―→	実行開始セクタのID情報
	C		←― R ―→	実行開始セクタのID情報
	C		←― N ―→	実行開始セクタのID情報
	C		←― EOT ―→	トラック上の最終セクタ番号
	C		←― GPL ―→	Gap3の長さ(VFO SYNCを含まず)
	C	W1	←― DTL ―→	処理すべきセクタ当りのデータ長
	E			データ転送
	R	R1	←― ST0 ―→	実行終了時のステータス0
	R		←― ST1 ―→	実行終了時のステータス1
	R		←― ST2 ―→	実行終了時のステータス2
	R		←― C ―→	実行終了セクタのID情報
	R		←― H ―→	実行終了セクタのID情報
	R		←― R ―→	実行終了セクタのID情報
	R	R1	←― N ―→	実行終了セクタのID情報

コマンド		B/WA$_0$	D_7 ―――――――――― D_0	備考
READ DELETED DATA				SK：Skip DAM
	C	W1	MT MF SK 0 1 1 0 0	
	C		× × × × × HD US1 US0	READ DATAと同じ
	C		←― C ―→	
	C		←― H ―→	
	C		←― R ―→	
	C		←― N ―→	
	C		←― EOT ―→	
	C		←― GPL ―→	
	C	W1	←― DTL ―→	
	E			データ転送
	R	R1	←― ST0 ―→	READ DATAと同じ
	R		←― ST1 ―→	
	R		←― ST2 ―→	
	R		←― C ―→	
	R		←― H ―→	
	R		←― R ―→	
	R	R1	←― N ―→	

コマンド		B/WA$_0$	D_7 ―――――――――― D_0	備考
READ ID				
	C	W1	0 MF 0 0 1 0 1 0	
	C	W1	× × × × × HD US1 US0	
	E			トラック上の最初に読んだエラーのないID情報をストアする。
	R	R1	←― ST0 ―→	READ DATAと同じ
	R		←― ST1 ―→	
	R		←― ST2 ―→	
	R		←― C ―→	E-Phaseで読んだID情報
	R		←― H ―→	
	R		←― R ―→	
	R	R1	←― N ―→	

コマンド		R/WA_0	D_7							D_0	備考
WRITE ID											
	C	W1	0	MF	0	0	1	1	0	1	
			×	×	×	×	×	HD	US1	US0	
			←			N				→	データ長／セクタ
			←			SC				→	セクタ数／トラック
			←			GPL				→	Gap3の長さ(VFO SYNCを含まず)
		W1	←			D				→	データ領域に書込むデータパターン
	E										トラック上のセクタ数分のID情報をメインシステムより転送する。
	R	R1	←			ST0				→	READ DATAと同じ（ただし，C,H,Rは無意味)
			←			ST1				→	
			←			ST2				→	
			←			C				→	
			←			H				→	
			←			R				→	
		R1	←			N				→	

コマンド		R/WA₀	D7 ——— D0	備考
WRITE DATA	C	W1	MT MF 0 0 0 1 0 1	READ DATAと同じ
			× × × × × HD US1 US0	
			← C →	
			← H →	
			← R →	
			← N →	
			← EOT →	
			← GPL →	
		W1	← DTL →	
	E			データ転送
	R	R1	← ST0 →	READ DATAと同じ
			← ST1 →	
			← ST2 →	
			← C →	
			← H →	
			← R →	
		R1	← N →	

コマンド		R/WA_0	D_7							D_0	備考
WRITE DELETED DATA	C	W1	MT	MF	0	0	1	0	0	1	READ DATAと同じ
			×	×	×	×	×	HD	US1	US0	
			←			C				→	
			←			H				→	
			←			R				→	
			←			N				→	
			←			EOT				→	
			←			GPL				→	
		W1	←			DTL				→	
	E										データ転送
	R	R1	←			ST 0				→	READ DATAと同じ
			←			ST 1				→	
			←			ST 2				→	
			←			C				→	
			←			H				→	
			←			R				→	
		R1	←			N				→	

コマンド	R/WA$_0$	D$_7$ ―――― D$_0$	備考
READ DIAGNOSTIC	W1	0 MF 0 0 0 0 1 0	READ DATAと同じ
C		× × × × × HD US1 US0	
		←―― C ――→	
		←―― H ――→	
		←―― R ――→	
		←―― N ――→	
		←―― EOT ――→	
		←―― GPL ――→	
	W1	←―― DTL ――→	
E			データ転送
R	R1	←―― ST0 ――→	READ DATAと同じ
		←―― ST1 ――→	
		←―― ST2 ――→	
		←―― C ――→	
		←―― H ――→	
		←―― R ――→	
	R1	←―― N ――→	

コマンド		R/WA0	D7 ―――――――――――――― D0	備　　　　考
SCAN EQUAL	C	W1	MT MF SK 1 0 0 0 1	READ DATAと同じ
			× × × × × HD US1 US0	
			← C →	
			← H →	
			← R →	
			← N →	
			← EOT →	
			← GPL →	
		W1	← STP →	比較すべきセクタのセクタ間隔1or2
	E			データ転送
	R	R1	← ST0 →	READ DATAと同じ
			← ST1 →	
			← ST2 →	
			← C →	
			← H →	
			← R →	
		R1	← N →	

コマンド	B/WA₀	D7 ———— D0	備考
SCAN LOW OR EQUAL	W1	MT MF SK 1 1 0 0 1	READ DATAと同じ
C		× × × × × HD US1 US0	
		←— C —→	
		←— H —→	
		←— R —→	
		←— N —→	
		←— EOT —→	
		←— GPL —→	
	W1	←— STP —→	比較すべきセクタのセクタ間隔1or2
E			データ転送
R	R1	←— ST0 —→	READ DATAと同じ
		←— ST1 —→	
		←— ST2 —→	
		←— C —→	
		←— H —→	
		←— R —→	
	R1	←— N —→	

コマンド	R/WA₀	D₇ ———— D₀	備　考
SCAN HIGH OR EQUAL	W1	MT MF SK 1 1 1 0 1	
C		× × × × × HD US1 US0	READ DATAと同じ
		←— C —→	
		←— H —→	
		←— R —→	
		←— N —→	
		←— EOT —→	
		←— GPL —→	
	W1	←— STP —→	比較すべきセクタのセクタ間隔1or2
E			データ転送
R	R1	←— ST 0 —→	READ DATAと同じ
		←— ST 1 —→	
		←— ST 2 —→	
		←— C —→	
		←— H —→	
		←— R —→	
	R1	←— N —→	

コマンド		R/WA₀	D₇ ——— D₀	備考
SEEK	C	W1	0 0 0 0 1 1 1 1	
		W1	× × × × × HD US1 US0	
			←—— NCN ——→	新シリンダ番号
	E			シーク動作
RECALI-BRATE	C	W1	0 0 0 0 0 1 1 1	
		W1	× × × × × 0 US1 US0	
	E			リカラブレイト動作
SENSE INT STATUS	C	W1	0 0 0 0 1 0 0 0	
	R	R1	←—— ST0 ——→	
		R1	←—— PCN ——→	コマンド終了時のシリンダ番号
SENSE DEVICE STATUS	C	W1	0 0 0 0 0 1 0 0	
		W1	× × × × × HD US1 US0	
	R	R1	←—— ST3 ——→	デバイスの状態
SPECIFY	C	W1	0 0 0 0 0 0 1 1	Step Rate Time, Head Unload Time
		W1	←—SRT—→ ←—HUT—→	Head Load Time, Non-DMA Mode
		W1	←—— HLT ——→ND	
Invalid	C	W1	←——その他のコード——→	
	R	R1	←—— STO ——→	ST0=80(16)

〔表5－5〕コマンドフェーズにおけるパラメータとバイト数

コ　マ　ン　ド	MT	MF	SK	HD	US	C	H	R	N
READ DATA	○	○	○	○	○	○	○	○	○
READ DELETED DATA	○	○	○	○	○	○	○	○	○
WRITE DATA	○	○		○	○	○	○	○	○
WRITE DELETED DATA	○	○		○	○	○	○	○	○
READ ID		○		○	○				
WRITE ID		○		○	○				○
READ DIAGNOSTIC		○		○	○	○	○	○	○
SCAN EQUAL	○	○	○	○	○	○	○	○	○
SCAN LOW OR EQUAL	○	○	○	○	○	○	○	○	○
SCAN HIGH OR EQUAL	○	○	○	○	○	○	○	○	○
SEEK				○	○				
RECALIBRATE				○	○				
SENSE INTERRUPT STATUS									
SENSE DEVICE STATUS				○	○				
SPECIFY									
INVALID									

EOT	GPL	DTL	SC	D	STP	NCN	SRT	HUT	HLT	ND	バイト数
○	○	○									9
○	○	○									9
○	○	○									9
○	○	○									9
											2
	○		○	○							6
○	○	○									9
○	○				○						9
○	○				○						9
○	○				○						9
						○					3
											2
											1
											2
							○	○	○	○	3
											—

〔表5-6〕リザルトフェーズにおけるパラメータとバイト数

コ　マ　ン　ド	ST0	ST1	ST2	ST3	C	H	R	N	PCN
READ DATA	○	○	○		○	○	○	○	
READ DELETED DATA	○	○	○		○	○	○	○	
WRITE DATA	○	○	○		○	○	○	○	
WRITE DELETED DATA	○	○	○		○	○	○	○	
READ ID	○	○	○		○	○	○	○	
WRITE ID	○	○	○		○	○	○	○	
READ DIAGNOSTIC	○	○	○		○	○	○	○	
SCAN EQUAL	○	○	○		○	○	○	○	
SCAN LOW OR EQUAL	○	○	○		○	○	○	○	
SCAN HIGH OR EQUAL	○	○	○		○	○	○	○	
SEEK									
RECALIBRATE									
SENSE INTERRUPT STATUS	○								○
SENSE DEVICE STATUS				○					
SPECIFY									
INVALID	○								

INT	RQM	バイト数
○		7
○		7
○		7
○		7
○		7
○		7
○		7
○		7
○		7
○		7
○		—
○		—
	○	2
	○	1
	○	—
	○	1

5－5　コマンドの機能

5－5－1　READ DATA

（1）　メインシステムから与えられるID情報(IDRと呼ぶ)で指定されるセクタのデータをディスクから読取り，1バイトごとにデータレジスタにセットします。メインシステム側はデータレジスタの内容をNon-DMAまたはDMAモードで読出します。

（2）　1セクタ読取り終了後メインシステムから読取り終了の指示（TERMINAL COUNT：以後TCと略す）がなければ，IDRのRバイトを更新し（R＋1→R），新しいIDRで指定されるセクタのデータを読取ります（マルチセクタ・リード）。ただし，Rバイトの更新はEOTバイトで指定される範囲内で行われます。

（3）　メインシステムからの読取り終了指示(TC)が，セクタの途中で発生された場合には，そのセクタの残りのデータはディスクから読取りますが，データレジスタにはセットせず，CRCのチェックのみを行います。CRCチェック終了後，コマンドを正常終了します。

INTまたはDRQが立上ってデータの転送要求が発生した直後から，標準フロッピィに対しては“1バイト・タイム－5μs”，ミニフロッピィに対しては“1バイト・タイム－10μs”内にTCを入力すると，そのINTまたはDRQで要求したデータ転送が終了した後はINTまたはDRQによるデータ転送要求は発生しなくなります。

ただし，セクタの最後のバイトの転送要求に対しては，INTまたはDRQが立上った直後から2バイト・タイム内にTCを入力することによって，次のセクタへのリード動作を止めることができます。

（4）　1回のコマンドの転送容量はIDRのNバイトとMTビットの指定により次の表に示すようになります。

この表で示される最終セクタの読取りが終った場合に，メインシステルからの読取り終了指示(TC)がなければ，ST1バイトのEN(End of Cylinder）をオンしてコマンドの実行を異常終了します。

（5）　1セクタのIDおよびデータを読取るごとにCRCバイトをチェックし，いずれかのセクタでCRCエラーが生じたら，ST1バイトのDE (Data Error) ビットをオンにしてコマンドの実行を異常終了します。ID領域でのCRCエラーに対してはDEビットのみオンにしますが，データ領域でのCRCエラーに対してはDEビットの他にST2

MT	MF	N(16)	転送容量(バイト)		最終セクタ
0	0	00	1～128×n	n＝1～26	ヘッド0のセクタ26
	1	01	1～256×n		または ヘッド1のセクタ26
1	0	00	1～128×n	n＝1～52	ヘッド1のセクタ26
	1	01	1～256×n		
0	0	01	1～256×n	n＝1～15	ヘッド0のセクタ15
	1	02	1～512×n		または ヘッド1のセクタ15
1	0	01	1～256×n	n＝1～30	ヘッド1のセクタ15
	1	02	1～512×n		
0	0	02	1～512×n	n＝1～8	ヘッド0のセクタ8
	1	03	1～1024×n		または ヘッド1のセクタ8
1	0	02	1～512×n	n＝1～16	ヘッド1のセクタ8
	1	03	1～1024×n		

備考　nの値はプログラマブルで，EOTの値で定義されます。

バイトのDD (Data Error in Data Field) ビットもオンします。

(6)　読取り開始時，HDビットで指定されるヘッドを選択した後，ヘッドのロード状態をチェックし，アンロード状態であるならばヘッドをロードし，ロード安定時間(SPECIFYコマンドのHLTで指定) 経過後，読取りを開始します。

(7)　コマンド実行終了後，ヘッドをすぐにはアンロード状態にせず，一定時間(SPECIFYコマンドのHUTで指定)経過後，アンロード状態にします。従ってこの時間以内に以前と同じデバイスに対してリードまたはライト系のコマンドが与えられたならば，ヘッドロードの安定時間待ちを節約することができます。

ただし，この時間以内に他のデバイスに対してコマンドが起動された場合，および以前と同じデバイスであってもシリンダが異なるSEEKコマンドが起動された場合には，以前のデバイスのヘッドはアンロード状態にします。またコマンドが起動された方のデバイスでは自動的にロード安定時間待ちが行なわれます。

(前述の(6)および(7)項はREAD DATAコマンドだけでなくリード／ライトを伴うコマンド総てに適用されます)

(8)　メインシステムからの読取り終了指示(TC)によりコマンド実行を正常終了したときには，Result Phaseで転送されるIDRにはMTビットおよびEOTバイトにより次表で示す値がセットされます。

MT	EOT(16)	最終バイトの転送に関係したセクタ	IDRの状態			
			C	H	R	N
0	1A 0F 08	ヘッド0のセクタ1〜25 〃 1〜14 〃 1〜7	もとのまま	もとのまま	R＋1	もとのまま
	1A 0F 08	ヘッド0のセクタ26 〃 15 〃 8	C＋1	もとのまま	R＝01 注	もとのまま
	1A 0F 08	ヘッド1のセクタ1〜25 〃 1〜14 〃 1〜7	もとのまま	もとのまま	R＋1	もとのまま
	1A 0F 08	ヘッド1のセクタ26 〃 15 〃 8	C＋1	もとのまま	R＝01 注	もとのまま
1	1A 0F 08	ヘッド0のセクタ1〜25 〃 1〜14 〃 1〜7	もとのまま	もとのまま	R＋1	もとのまま
	1A 0F 08	ヘッド0のセクタ26 〃 15 〃 8	もとのまま	下1ビット を反転する	R＝01 注	もとのまま
	1A 0F 08	ヘッド1のセクタ1〜25 〃 1〜14 〃 1〜7	もとのまま	もとのまま	R＋1	もとのまま
	1A 0F 08	ヘッド1のセクタ26 〃 15 〃 8	C＋1	下1ビット を反転する	R＝01 注	もとのまま

注　ヘッドの選択状態(ST0のHDビットも含む)は最終バイトの転送に関係したセクタの属するトラックのままです。

(9)　Deleted Data Address Mark を検出したときには，ST2 バイトの CM (Control Mark)ビットをオンにします。このときコマンドの SK ビットの内容により二通りの処理が行なわれます。SK ビットが 0 のときはそのセクタを読取り後コマンドの実行を正常終了します。Result Phase で転送される CHRN は Deleted Data Address Mark を検出したセクタの値になります。SK ビットが 1 のときはそのセクタをスキップして次のセクタを処理します。

(10)　セクタのデータをメインシステムに転送しているとき，一定時間内 (標準フロッピィに対して FM モードの場合 27μS，MFM モードの場合 13μS 以内；ミニフロッピィに対して FM モードの場合 54μS，MFM モードの場合 26μS 以内)にメインシステ

ムのサービスが行われなかった場合には，そのセクタを読取り後，ST1バイトのOR (Over Run) ビットをオンにしてコマンドの実行を異常終了します。

(11) IDRのNバイトが00のときには，DTLバイトによってセクタ当りの処理すべきデータ長が指定されます。DTLがセクタのデータの途中までを指定してある場合には，そのセクタの残りのデータはディスクから読取りますが，データレジスタにはセットせず，CRCチェックのみを行ないます。

DTL＝$01_{(16)}$が1バイト／セクタを，$80_{(16)}$が128バイト／セクタを指定します。この機能を用いない場合にはDTLバイトは$FF_{(16)}$とします。

IDRのNバイトが00以外のときにはDTLバイトは意味を持ちません。

なお，マルチセクタ・リードを行なうか否かは，メインシステムからの読取り終了指示 (TC) に従います。

(12) コマンド実行開始時及びIDアドレスマーク・サーチ時，デバイスがノットレディ状態であった場合，ST0バイトのNR (Not Ready) ビットをオンにしてコマンドの実行を異常終了します。

(13) IDRで指定されるセクタがクンデックス・マークを2回検出するまでに見つからない場合は，ST1バイトのND (No Data) ビットをオンにしてコマンドの実行を異常終了します。

このとき読取ったIDのCバイトがIDRのCバイトと一致せず，かつ$FF_{(16)}$でなければ，NDの他にST2バイトのNC (No Cylinder) ビットもオンにします。また，IDのCバイトがIDRのCバイトと一致せず，かつ$FF_{(16)}$であれば，NDの他にST2バイトのBC (Bad Cylinder) ビットもオンにします。

なお，IDRで指定されるセクタがインデックス・マークを2回検出するまでに見つからず，かつこの間にIDアドレスマークが1回も検出されなかった場合には，NDの代りにST1のMA (Missing Address Mark) をオンにして異常終了します。

(14) IDRで指定されるセクタのIDを検出後，データ部のアドレスマーク (Data Address Mark or Deleted Data Address Mark) を一定時間 (1ms ／ 8MHz) 以内に検出できなかった場合には，ST1のMAビットとともに，ST2のMD (Missing Address Mark in Data Field) ビットをオンにしてコマンドの実行を異常終了します。

(15) 各パラメータとセクタフォーマットの関係を次の表に示します。

GPLはマルチセクタ・アクセス時，データ部と次のセクタのID部の間のSplice Point (不連続点) を避けるために必要なパラメータです。

フォーマット ＼ パラメータ		N(16)	EOT(16)	GPL(16)	備　　　考
FMモード	１２８バイト／セクタ	０ ０	１ A	０ ７	IBMディスケット1
	２５６	０ １	０ F	０ E	IBMディスケット2
	５１２	０ ２	０ ８	１ B	
	１０２４	０ ３	０ ４	未定	
	２０４８	０ ４	０ ２	未定	
	４０９６	０ ５	０ １	未定	
MFMモード	２５６バイト／セクタ	０ １	１ A	０ E	IBMディスケット2D
	５１２	０ ２	０ F	１ B	
	１０２４	０ ３	０ ８	３ ５	IBMディスケット2D
	２０４８	０ ４	０ ４	未定	
	４０９６	０ ５	０ ２	未定	
	８１９２	０ ６	０ １	未定	

備考　ＧＰＬはプログラマブルです。

フォーマット ＼ パラメータ				N(16)	EOT(16)	GPL(16)
ECMA66 (片面)	FM	トラック00	128バイト／セクタ	０ ０	１ ０	０ ７
		トラック00以外	256　〃	０ １	０ ９	０ E
ECMA70 (両面)	FM	サイド０のトラック00	128　〃	０ ０	１ ０	０ ７
	MFM	サイド０のトラック00以外	256　〃	０ １	１ ０	０ E

備考　GPLはプログラマブルです。

5－5－2　READ DELETED DATA

（1）　セクタのデータフィールドのData Address MarkとDeleted Data Address Markを互いに読変えたREAD DATA　コマンドと同じです。

5－5－3　WRITE DATA

（1）　IDRで指定されるセクタのデータフィールドへ，メインシステムから1バイトごとにNon-DMAまたはDMAモードでデータレジスタに転送されてくるデータを書込みます。

（2）　1セクタ書込み終了後，メインシステムから書込み終了の指示（TC）がなければ，IDRのRバイトを更新し（R＋1→R），新しいIDRで指定されるセクタにデータを書込みます（マルチセクタ・ライト）。ただし，Rバイトの更新はEOTバイトの範囲内で行なわれます。

（3）　メインシステムからの書込み終了（TC）がセクタのデータの途中で指示された場合には，そのセクタの残りのバイトには$00_{(16)}$を書込んでコマンドを正常終了します。

INTまたはDRQが立上ってデータの転送要求が発生した直後から，標準フロッピィに対しては“1バイト・タイム－6μS”，ミニフロッピィに対しては“1バイト・タイム－12μS”内にTCを入力すると，そのINTまたはDRQによるデータ転送が終了した後はINTまたはDRQによるデータ転送要求は発生しなくなります。

ただし，セクタの最後のバイトの転送要求に対しては，INTまたはDRQが立上った直後から2バイト・タイム内にTCを入力することによって，次のセクタに対するライト動作を止めることができます。

（4）　1セクタのIDを読取るごとにCRCバイトをチェックし，いずれかのセクタでCRCエラーが生じたら，ST1バイトのDEビットをオンにしてコマンドの実行を異常終了します。

（5）　セクタのデータフィールドへ書込むデータをメインシステムから転送しているとき，一定時間内(標準フロッピィに対してFMモードの場合は31μS，MFMモードの場合は15μS以内；ミニフロッピィに対してFMモードの場合は62μS，MFMモードの場合は30μS以内)にメインシステムのサービスが行われなかった場合には，そのセクタを書込み後ST1バイトのORビットをオンにしてコマンドの実行を異常終了します。

（6）　IDRのNバイトが$00_{(16)}$のときには，DTLバイトによってセクタ当りの処理すべきデータ長が指定されます。DTLがセクタのデータの途中までを指定してある場合に

は，そのセクタの残りのデータには$00_{(16)}$を書込みます。

DTL＝$01_{(16)}$が１バイト／セクタを，$80_{(16)}$が128バイト／セクタを指定します。この機能を用いない場合にはDTLバイトは$FF_{(16)}$とします。

IDRのNバイトが$00_{(16)}$以外のときにはDTLバイトは意味を持ちません。

なお，マルチセクタ・ライトを行うか否かは，メインシステムからの書込み終了指示（TC）に従います。

（７） データ部のCRCバイトの後に，ギャップを１バイト書込み後，FAULT信号がオンの場合，ST0バイトのEC（Equipment Check）ビットをオンにしてコマンドの実行を異常終了します。

（８） コマンド実行開始時，WRITE　PROTECTED信号がオンの場合，ST1バイトのNW（Not Writable）ビットをオンにしてコマンド実行を異常終了します。

（９） READ DATA　コマンドの（４），（６），（７），（８），(12)，(13)と同じ。

5－5－4　WRITE DELETED DATA

（１） セクタのデータフィールドのData　Address　Markの代りに，Deleted　Data　Address Markを書込む事を除いてWRITE DATAコマンドと同じです。

5－5－5　READ ID

（１） 選択したトラック上で最初に検出したエラーでないIDをIDRに格納します。格納したIDRの情報はResult Phaseでデータレジスタにセットし，メインシステムはその内容を読取ります。

（２） 最大１トラック分読取りを行ってもエラーでないIDがインデックス・マークを２回検出するまで見つからない場合は，ST1バイトのND（No Data）ビットをオンにしてコマンドの実行を異常終了します。

なお，エラーでないIDがインデックス・マークを２回検出するまでに見つからず，かつIDアドレスマークが１回も検出されなかったときには，NDビットの代りにST1バイトのMA（Missing Address Mark）をオンにして異常終了します。

（３） READ DATA　コマンドの（６），（７），(12)と同じ。

5－5－6　WRITE ID（Format Write）

（１） Command Phaseでメインシステムより与えられるセクタ長(N)，トラック当りのセクタ数（SC），ギャップ長（GPL），データ部に書込むデータパターン（D）などに従ってフォーマット書込みを行ないます。

(2) セクタのID部にデータを書込むときには，メインシステムから転送されるデータ（C，H，R，Nの4バイト×セクタ数）を書込みます。従って，メインシステムから転送するデータによってセクタシーケンスIDや不良シリンダIDなども書込むことができます。

(3) 各セクタのデータ部には，Dバイトのデータを Nバイトで指定される長さ分だけ繰り返し書込みます。

(4) 1セクタ書込み終了後，メインシステムから書込み終了の指示（TC）がなく，かつSCバイトで指定されたセクタ数分のセクタ・フォーマット書込みが終了していなければIDRのRバイトを更新し（R＋1→R)，セクタ・フォーマット書込みを続行します。ただし，Rバイトにはコマンド実行開始時に初期値$01_{(16)}$をセットします。

(5) 1セクタ書込み終了後，メインシステムから書込み終了指示(TC)があるか，SCバイトで指定されたセクタ数分のセクタ・フォーマット書込みが終了すると次のインデックス・パルスまでギャップを書込みます。

(6) 書込み終了時，FAULT信号がオンの場合，ST0バイトのEC(Equipment Check)ビットをオンにしてコマンドを異常終了します。

(7) セクタのID部へ書込むデータをメインシステムから転送しているとき，一定時間内（標準フロッピィに対してFMモードの場合31μS，MFMモードの場合15μS以内；ミニフロッピィに対してFMモードの場合62μS，MFMモードの場合30μS以内）にメインシステムのサービスが行われなかった場合には，トラックの書込み後、ST1バイトのOR（Over Run）ビットをオンにしてコマンドを異常終了します。。

(8) コマンド実行開始時，デバイスがノットレディ状態であた場合，ST0バイトのNR（Not Ready）ビットをオンにしてコマンドを異常終了します。

(9) READ DATA コマンドの(6)，(7)と同じ。

(10) WRITE DATA コマンドの(8)と同じ。

(11) 各パラメータとセクタ・フォーマットの関係を次の表に示します。

フォーマット ＼ パラメータ		N(16)	EOT(16)	GPL(16)	備考
FMモード	128バイト／セクタ	00	1A	1B	IBMディスケット1
	256	01	0F	2A	IBMディスケット2
	512	02	08	3A	
	1024	03	04	未定	
	2048	04	02	未定	
	4096	05	01	未定	
MFMモード	256バイト／セクタ	01	1A	36	IBMディスケット2D
	512	02	0F	54	
	1024	03	08	74	IBMディスケット2D
	2048	04	04	未定	
	4096	05	02	未定	
	8192	06	01	未定	

備考　GPLはプログラマブルです。

フォーマット ＼ パラメータ				N(16)	EOT(16)	GPL(16)
ECMA66 (片面)	FM	トラック00	128バイト／セクタ	00	10	18
		トラック00以外	256 〃	01	09	26
ECMA70 (両面)	FM	サイド0のトラック00	128 〃	00	10	18
	MFM	サイド0のトラック00以外	256 〃	01	10	32

備考　GPLはプログラマブルです。

5－5－7　READ DIAGNOSTIC

（1）　インデックス・マークの直後のセクタから読取り開始しますが次に示す相違点を除いてREAD DATAコマンドと同じです。

- IDまたはデータ部のCRCエラーが検出されても読取りを続けます。
- 読取ったIDとIDRの比較はしますが，それらが異なっていてもST1バイ

トのND (No Data) ビットをオンにするだけで，そのセクタのデータは処理します。

- マルチトラック指定，スキップ指定はありません。
- デリーテッド・データ・マークのセクタを検出してもコマンド実行終了条件にはなりません。

5－5－8　SCAN EQUAL／SCAN LOW OR EQUAL／SCAN HIGH OR EQUAL

（1） IDRで指定されるセクタから開始し，セクタのデータをディスクから読取り，1バイトごとにメインシステムから送られてくるデータと比較し，条件にあったセクタを検出します。

（2） 1セクタの比較で条件に合うセクタが検出できないときには，IDRのRを更新し（R＋STP→R：STPは1または2)，新しいセクタについて比較動作を行ないます。したがってメインシステムからは条件に合ったセクタが検出されるまで，同じデータ群をくり返し送り込む必要があります。

（3） 各セクタの最終バイトを転送したときには必ずTCを入力して下さい。各セクタの全バイトを比較する必要のないときは，セクタの途中でTCを入力し，セクタの後半の比較を行わずに比較を終了させ，それまでのバイトに対して一致，大，小の比較を見ることができます。

（4） メインシステムからのデータが$FF_{(16)}$のときは，そのデータについては比較を行ないません。

（5） 比較条件は次の通りです。

- SCAN EQUALでは，比較すべきバイト群が等しいか否かをチェックし，等しいときコマンドの実行を正常終了します。
- SCAN LOW OR EQUALでは，比較すべきバイト群について，ディスクの方がメインシステムのデータより小さいか等しいときコマンドの実行を正常終了します。
- SCAN HIGH OR EQUALでは，比較すべきバイト群についてディスクの方がメインシステムのデータより大きいか等しいときコマンドの実行を正常終了します。

一致，大，小の比較は最初に比較するバイトを最上位として行ないます。

		MSB					
ディスク	DAM	0 1	4 1	5 0	6 5	7 3	2 1
		↕	↕	↕	↕	↕	↕
メイン		0 1	4 1	F F	6 5	7 3	2 1

ディスク＝メイン

		MSB					
ディスク	DAM	0 1	4 1	5 0	6 5	7 3	2 1
		0↕	4↕	↕	↕	↕	↕
メイン		0 1	4 1	5 6	4 1	7 3	0 0

ディスク＜メイン

		MSB					
ディスク	DAM	0 1	4 1	5 0	6 5	7 3	2 1
		↕	↕	↕	↕	↕	↕
メイン		0 1	4 1	5 0	4 7	9 0	F F

ディスク＜メイン

(6) 比較条件のうち等しい条件を満足したときは，ST2バイトのSH(Scan Equal Hit)をオンにしてコマンドの実行を正常終了します。

最終セクタまで比較をしても条件を満足するセクタが検出されなかったときは，ST2バイトのSN (Scan Not Satisfied) ビットをオンにしてコマンドの実行を正常終了します。

(7) Deleted Data Address Markのセクタを検出したときにはST2バイトのCM (Control Mark) ビットをオンにします。このときSKビットの内容により二通りの処理が行われます。

SKビットが0の場合には，そのセクタを最終セクタとみなして動作します。SKビットが1の場合には，そのセクタをスキップして次のセクタを処理します。

(8) Command Phaseで転送されるSTPバイトは$01_{(16)}$または$02_{(16)}$が可能ですが，$02_{(16)}$

のときにはCommand Phaseで転送されるRバイトとEOTバイトは次式を満足していなければなりません。

R＋2 (n－1)＝EOT　　　n：処理したセクタ数

(9) 比較すべきデータをメインシステムから転送しているとき，一定時間内（標準フロッピィに対してFMモードの場合27μS，MFMモードの場合13μS以内；ミニフロッピィに対してFMモードの場合54μS，MFMモードの場合26μS以内）にメインシステムのサービスが行われなかった場合には，そのセクタの比較動作終了後，ST1バイトのOR（Over Run）ビットをオンにしてコマンドの実行を異常終了します。

(10) READ DATAコマンドの(5)，(6)，(7)，(12)，(13)，(14)と同じ。

5－5－9 SEEK

(1) Command Phaseで転送されるNCN（New Cylinder Number）バイトをシリンダ物理番号とみなして，目的シリンダまでシークさせます。

(2) シーク動作としては，現在のリード／ライト・ヘッドの位置を記憶しているPCN（Present Cylinder Mumber）バイトとNCNバイトとを比較します。比較結果が異なっている場合次の二通りの動作が行なわれます。

① NCN＞PCNのとき
DIRECTION信号を1にしてSTEP信号を出力し，PCNをインクリメント（PCN＋1→PCN）します。

② NCN＜PCNのとき
DIRECTION信号を0にしてSTEP信号を出力し，PCNをデクリメント（PCN－1→PCN）します。

その後はSPECIFYコマンドで指定されるステップ時間ごとに以上の動作を比較結果が等しくなるまで繰り返します。

比較結果が等しい場合，ST0バイトのSE（Seek End）をオンにしてコマンドの実行を正常終了します。ただし，ST0バイトの内容はSENSE INTERRUPT STATUSコマンドのResult Phaseの中で処理されます。

(3) SEEKコマンドはCommand PhaseではFDC Busy状態ですが，Execution PhaseではNon-Busy状態であるため，このときに他のデバイスに対するSEEKまたはRECALIBRATEコマンドを受付けることができ，デバイス4台までのシークの同時動作が可能です。

FDCがNon－Busy状態であっても，いずれかのデバイスがBusy状態のときには，FDCにリード／ライト系のコマンドを実行させてはいけません。

（4） SEEK コマンド開始時及びシーク動作を実行中，デバイスがノットレディ状態であった場合には，ST0 バイトの SE ビットと NR ビットをオンにしてコマンドを異常終了します。

5－5－10 RECALIBRATE (Return to cylinder 0)

（1） シリンダ物理番号 0 (デバイスからの TRACK0 信号を検出するまで) へシークさせます。

（2） シーク動作としては，デバイスの現在のリード／ライト・ヘッドの位置を記憶している PCN バイトをクリアした後，デバイスからの TRACK0 信号をチェックします。

TRACK0 信号が 0 の場合，DIRECTION 信号を 0 にして STEP 信号を出します。その後は SPECIFY コマンドで指定されるステップ時間ごとに前述の動作を TRACK0 信号が 1 になるまで繰り返します。

TRACK0 信号が 1 になったら，ST0 バイトの SE (Seek End) ビットをオンにしてコマンドの実行を正常終了します。

（3） （2）項の動作を77回（μPD765A，745）または255回（μPD7265）繰り返しても TRACK0 信号が 1 にならない場合，ST0 バイトの SE ビットと EC（Equipment Check）ビットをオンにしてコマンドを異常終了します。

（4） SEEK コマンドの（3），（4）と同じ。

5－5－11 SENSE INTERRUPT STATUS

（1） SEEK または RECALIBRATE コマンドによるシーク動作終了時のリザルトステータス，またはコマンド実行中なないときのデバイスの状態遷移（Not Ready から Ready へ，あるいは Ready から Not Ready へ）時のリザルトステータスをデータレジスタにセットし，RQM をオンにしてメインシステムに対してそれらの引取り要求をします。

（2） PCN バイトは，シーク動作終了時のシリンダ物理番号を示します。状態遷移時の PCN バイトは意味を持ちません。

（3） シーク動作終了または状態遷移がいずれのデバイスにも発生していないときこのコマンドは Invalid コマンドとして処理されます（INVALID コマンドの項を参照してください）。

5－5－12 SPECIFY

（1） FDC の各種時間用タイマおよび FDC の動作モードの初期値を FDC にセットしま

す。

（2） HUT（Head Unlodad Time）は，リード／ライトコマンド実行後アンロード状態にするまでの時間を指定します。標準フロッピィに対して16mS単位で16mSから240mSまで，ミニフロッピィに対して32mS単位で32mSから480mSまで指定します。

HUT	時間（ms）
0	禁　止
1	16／32
2	32／64
3	48／96
4	64／128
5	80／160
6	96／192
7	112／224

HUT	時間（ms）
8	128／256
9	144／288
A	160／320
B	176／352
C	192／384
D	208／416
E	224／448
F	240／480

（3） SRT（Step Rate Time）は，シーク動作時デバイスへ送るSTEP信号の間隔時間を指定します。標準フロッピィに対して1mS単位で1mS（コードF(16)に対応）から16mS（コード0に対応）まで，ミニフロッピィに対して2mS単位で2mSから32mSまで指定できます。

SRT	時間（ms）
0	16／32
1	15／30
2	14／28
3	13／26
4	12／24
5	11／22
6	10／20
7	9／18

SRT	時間（ms）
8	8／16
9	7／14
A	6／12
B	5／10
C	4／8
D	3／6
E	2／4
F	1／2

この表の値は標準値で，最小値は表のそれぞれの値から標準フロッピィ時(8MHz)は1mS，ミニフロッピィ時（4MHz）は2mSを引いてください。

（4） HLT（Head Load Time）は，リード／ライト・コマンドの実行開始時，ヘッドがアンロード状態であればロードさせますが，ロードした後ヘッドが安定状態になるま

での待ち時間を指定します。標準フロッピィに対して2mS単位で2mSから254mS，ミニフロッピィに対して4mS単位で4mSから508mSまで指定できます。

(5) ND (Non DMA Mode) ビットは，リード／ライト・コマンド実行時，Execution

HLT	時間 (ms)
0 0	禁　止
0 1	2／4
0 2	4／8
0 3	6／12
0 4	8／16
0 5	10／20
0 6	12／24
0 7	14／28

HLT	時間 (ms)
0 8	16／32
0 9	18／36
7 D	250／500
7 E	252／504
7 F	254／508

Phaseんにおいてメインシステムとのデータ転送モードを指定します。1のときNon DMAモード，0のときDMAモードを指定します。

5－5－13　SENSE DEVICE STATUS

(1) FDCへ入力されるデバイス信号の状態をST3バイトにセットし，それをデータレジスタにセットしてRQMビットをオンにし，メインシステムに対して引取り要求をします。

5－5－14　INVALID

(1) Command Phaseにおいて定義されていないコマンドコードが与えられたとき，またはSENSE INTERRUPT STATUSコマンド実行時，SEEK/RECALIBRATEコマンドの終了INT要求および状態遷移によるINT要求のいずれも起こってなかった場合，ST0バイトのInterruption Codeんに$10_{(16)}$ (IC:Invalid Commoand)，その他の部んにはすべて0をセットします。その後ST0バイトの内容をデータレジスタに移してステータスレジスタのRQMビットをオンにし，メインシステムに対して引取り要求をします。

検印省略　　NDC 548

フロッピーディスク
プロテクト活用ハンドブック

昭和59年12月24日　　発　行

定価　1800円

著　　者　　エリートソフト
発行者　　小　川　茂　男
発行所　　株式会社誠文堂新光社
東京都千代田区神田錦町1－5－5
郵便番号 101
電話　東京　(292)　1 2 1 1
振替口座　東京　7－6 2 9 4
印刷・広研印刷株式会社
製本・岡嶋製本工業

ISBN4-416-18448-4 C2055

フロッピーディスク
プロテクト活用ハンドブック

第１章　標準的物理フォーマット

第２章　基本的Aプロテクト

第３章　より高度なプロテクト

第４章　同一FDCでは外せないプロテクト

第５章　プロテクトを持つプログラムの開発

付録１　MB8876A，MB887Aの仕様書

付録２　μPD765A/745/7265の仕様書

定価1800円

誠文堂新光社

ISBN4-416-18448-4 C2055 ¥1800E